Graph-Theoretic Algorithms for the Alternating Trilinear Form Equivalence Problem

Ward Beullens

IBM Research, Zurich, Switzerland
wbe@zurich.ibm.com

Abstract. At Eurocrypt'22 Tang, Duong, Joux, Plantard, Qiao, and Susilo proposed a digital signature algorithm based on the hardness of the isomorphism problem of alternating trilinear forms. They propose three concrete parameters in dimensions 9,10, and 11 respectively. We give new heuristic algorithms that solve this problem more efficiently. With our new algorithms, the first parameter set can be broken in less than a day on a laptop. For the second parameter set, we show there is a 2^{-17} fraction of the public keys that can also be broken in less than a day. We do not break the third parameter set in practice, but we claim it falls short of the target security level of 128 bits.

1 Introduction

We are interested in the Alternating Trilinear Form Equivalence (ATFE) problem, which is defined as follows:

Definition 1 (ATFE). *Given a pair of equivalent alternating trilinear forms $\phi_1, \phi_2 \in ATF(\mathbb{F}_q^n)$, the ATFE problem asks to find an equivalence $S \in GL(\mathbb{F}_q^n)$ such that $\phi_2(x, y, z) = \phi_1(Sx, Sy, Sz)$.*

This problem was shown to be complete for the Tensor Isomorphism complexity class (TI) [21,13], and believed to be hard on average, even for quantum algorithms. Therefore, the authors of [20] argued that ATFE is a good basis for post-quantum cryptography, and they proposed a digital signature algorithm based on the hardness of the ATFE problem. They propose three concrete instances of the ATFE problem, using alternating trilinear forms in dimensions 9, 10, and 11 over fields of order $524287, 131071$, and 65521 respectively. Their signature scheme fits in a family of signature schemes based on the GMW zero-knowledge proof protocol for Graph isomorphisms [12], which has been generalized to many isomorphism problems, resulting in isogeny-based [9,11,4], multivariate [18], and code-based [5] signature algorithms. Some of these signature schemes have been broken in practice because the isomorphism problems turned out to be easier to solve than expected [6,2].

* Ward Beullens holds Junior Post-Doctoral fellowship 1S95620N from the Research Foundation Flanders (FWO).

After the work of [20], the ATFE problem has been used to construct ring signatures independently by D'Alconzo and Gangemi [10] and Chen *et al.* [8], both using the framework of Beullens *et al.* [3]. Leroux and Roméas [16] construct an updateable encryption scheme based on the hardness ATFE. To have confidence in the security of these cryptographic systems, and to be able to pick secure parameter sets, it is important to investigate the concrete hardness of the ATFE problem.

Contributions. In this paper we give new heuristic algorithms for solving the ATFE problem. Since our main motivation is to break the cryptosystem proposed by [20] we focus on dimensions $9, 10$ and 11. Our results are summarized in Table 1. Our new algorithms are more efficient than existing algorithms and can solve the ATFE problems proposed by [20] in dimension 9 in at most 4 hours. We also show that a $1/q$ fraction of the proposed ATFE problems in dimension 10 can be solved in practice in approximately 1.5 hours. We do not break the proposed parameter set in dimension 11, but we estimate that it can be broken in 2^{60} core-hours on modern CPUs, which means that the signature scheme is less secure than the target security level of 128 bits.

Implementations of some (parts of) of our algorithms, including a complete implementation of our $O(q)$ algorithm for solving the ATFE problem in dimension 9, and the algorithms to reproduce Table 2 are publicly available through the following link:

`https://github.com/WardBeullens/BreakingATFE` .

Table 1. Algorithms for solving the ATFE problem in dimensions $9, 10$ and 11.

Dimension n	Algorithm	Complexity (# field ops.)	Note
9	Tang *et al.* [20]	$O(q^7)$	At rank $R = 6$
	Section 5.3	$O(q^2)$	At rank $R = 4$
	Section 7	$O(q)$	Practical for $q = 524287$.
10	Tang *et al.* [20]	$O(q^7)$	At rank $R = 6$
	Section 5.3	$O(q^6)$	At rank $R = 6$
	Section 6	$O(1)$	Works only for $1/q$-fraction of instances. Practical for $q = 131071$
11	Tang *et al.* [20]	$O(q^9)$	At rank $R = 6$
	Section 5.3	$O(q^4)$	At rank $R = 6$

2 Preliminaries

Let $\mathbb{F}_q$ be a prime field of odd characteristic. We denote by $\mathcal{S}(q, n)$ be the space of skew-symmetric n-by-n matrices over $\mathbb{F}_q$.

Projective space and projective frames. Let V be an n-dimensional vector space over a field $\mathbb{F}_q$. This defines a projective space $\mathbb{P}(V) := V \setminus \{0\} \mod \sim$, where $\mathbf{u} \sim \mathbf{v}$ if there exists $\alpha \in \mathbb{F}_q$ such that $\mathbf{u} = \alpha\mathbf{v}$. We denote the equivalence class of $\mathbf{u} \in V$ as $\bar{\mathbf{u}}$. It is well known that if $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is a basis for V, we can uniquely represent vectors $\mathbf{v}$ in V with n coordinates $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$, such that $\mathbf{v} = \sum_{i=1}^{n} \alpha_i \mathbf{b}_i$. Analogously, we call a sequence of $n+1$ projective points $\bar{\mathbf{b}}_1, \ldots, \bar{\mathbf{b}}_{n+1} \in \mathbb{P}(V)$ a projective frame (or projective basis) if no n of them are contained in a hyperplane of $\mathbb{P}(V)$. One can always pick representatives for the projective points in a frame such that $\sum_{i=1}^{n} \mathbf{b}_i = \mathbf{b}_{n+1}$, and this choice of representatives is unique up to multiplication by a scalar in $\mathbb{F}_q$, so we can describe any projective point $\bar{\mathbf{v}} \in \mathbb{P}(V)$ with homogeneous coordinates α_i such that $\sum_i \alpha_i \mathbf{b}_i = \mathbf{v}$, and this representation is unique up to multiplication by a scalar in $\mathbb{F}_q$.

Alternating trilinear forms. A trilinear form on V is a function $\phi : V \times V \times V \to \mathbb{F}_q$ that is linear in each of its three arguments, e.g., $\phi(\alpha\mathbf{u} + \mathbf{u}', \mathbf{v}, \mathbf{w}) = \alpha\phi(\mathbf{u}, \mathbf{v}, \mathbf{w}) + \phi(\mathbf{u}', \mathbf{v}, \mathbf{w})$ for all $\mathbf{u}, \mathbf{u}', \mathbf{v}, \mathbf{w} \in V$ and all $\alpha \in \mathbb{F}_q$. We say a trilinear form ϕ is *alternating* if $\phi(\mathbf{u}, \mathbf{v}, \mathbf{w}) = 0$ when at least two of the three inputs are equal. Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be a basis for V, and let $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ be the corresponding dual basis. The alternating trilinear forms make up a $\mathbb{F}_q$-vectorspace, and a basis is given by trilinear forms $\mathbf{b}_i^* \wedge \mathbf{b}_j^* \wedge \mathbf{b}_k^*$, where $1 \leq i < j < k \leq n$, which are defined as

$$(\mathbf{b}_i^* \wedge \mathbf{b}_j^* \wedge \mathbf{b}_k^*)(\mathbf{u}, \mathbf{v}, \mathbf{w}) := \begin{vmatrix} \mathbf{b}_i^*(\mathbf{u}) & \mathbf{b}_j^*(\mathbf{u}) & \mathbf{b}_k^*(\mathbf{u}) \\ \mathbf{b}_i^*(\mathbf{v}) & \mathbf{b}_j^*(\mathbf{v}) & \mathbf{b}_k^*(\mathbf{v}) \\ \mathbf{b}_i^*(\mathbf{w}) & \mathbf{b}_j^*(\mathbf{w}) & \mathbf{b}_k^*(\mathbf{w}) \end{vmatrix},$$

which implies that the space of alternating trilinear forms has dimension $\binom{n}{3}$. We denote the space of alternating trilinear forms on V by $\mathrm{ATF}(V)$.

Radicals. If ϕ is an alternating trilinear form and $\mathbf{u} \in V$, then we denote by $\phi_{\mathbf{u}}$ the alternating bilinear form $\phi_{\mathbf{u}}(\mathbf{v}, \mathbf{w}) := \phi(\mathbf{u}, \mathbf{v}, \mathbf{w})$. Similarly, for $\mathbf{u}, \mathbf{v} \in V$ we define the linear form $\phi_{\mathbf{u},\mathbf{v}}(\mathbf{w}) := \phi(\mathbf{u}, \mathbf{v}, \mathbf{w})$. We define the *radical* $\mathrm{Rad}(\phi)$ of a trilinear form ϕ as the space $\{\mathbf{x} \,|\, \phi_{\mathbf{x}} = 0\}$. We say ϕ is *non-degenerate* if $\mathrm{Rad}(\phi)$ is trivial. For a vector $\mathbf{u} \in V$ we define $\mathrm{Rad}_\phi(\mathbf{u}) := \{\mathbf{x} \,|\, \phi_{\mathbf{u},\mathbf{x}} = 0\}$, and we say the *rank* of $\mathbf{u}$ (with respect to ϕ) is the rank of $\phi_{\mathbf{u}}$, which equals the codimension of $\mathrm{Rad}_\phi(\mathbf{u})$. Note that $\mathrm{Rad}_\phi(\mathbf{u})$ always contains $\langle\mathbf{u}\rangle$ (because ϕ is alternating), so the rank of $\mathbf{u}$ is always strictly smaller than n. Moreover, $\mathrm{rank}(\mathbf{u})$ is always even.

Solving the MinRank problem. Our algorithms for solving the equivalence problem of alternating trilinear forms will use a subroutine to solve the MinRank problem. Given k matrices $\mathbf{M}_1, \ldots, \mathbf{M}_k \in \mathbb{F}_q^{n \times n}$ and a target rank r, the MinRank problem asks to find a linear combination $\sum_i \alpha_i \mathbf{M}_i$ with rank at most r. This problem has been studied relatively well because it is relevant for the security of many cryptosystems. We will use the MinRank-solving algorithm of [1]. Let $\mathbf{M} = \sum_i \alpha_i \mathbf{M}_i$ be the rank-r linear combination that we are looking for, and let $\mathbf{M} = \mathbf{HC}$ with $\mathbf{H} \in \mathbb{F}_q^{n \times r}$ and $\mathbf{C} \in \mathbb{F}_q^{r \times n}$ be a rank-decomposition of $\mathbf{M}$. If

$\mathbf{M}^{(j)} = \sum \alpha_i \mathbf{M}_i^{(j)}$ is the j-th row of $\mathbf{M}$, then the matrix

$$\begin{pmatrix} \mathbf{M}^{(j)} \\ C \end{pmatrix}$$

obtained by adding $\mathbf{M}^{(j)}$ on top of $\mathbf{C}$ has rank r, so all $\binom{n}{r+1}$ of its $(r+1)$-by-$(r+1)$ minors vanish. The idea behind the algorithm of [1] is that after doing a cofactor expansion along the top row we get a bilinear equation in the coefficients α_i and the r-by-r minors of C of which there are $\binom{n}{r}$. We then try to solve a system of equations, whose variables are the α_i and the minors of C, by linearization. We expect this system of equations to have a unique solution if the number of equations $n\binom{n}{r+1}$ is at least the number of monomials $k\binom{n}{r}$, which happens if $n(n-r) \geq (r+1)k$ and if the MinRank problem has a unique solution. The bottleneck of the algorithm is doing linear algebra on a square matrix of size $k\binom{n}{r}$, which would take $O(k^3\binom{n}{r}^3)$ field operations with Gaussian elimination. However, we can take advantage of the sparsity of the matrix and solve the system with the Wiedemann solver in $O(k^3 r\binom{n}{r}^2)$ field operations instead [1].

3 The graph of alternating trilinear forms

Let $\phi \in \mathrm{ATF}(V)$ be an alternating trilinear form, then we can define the graph G_ϕ, as the undirected graph with vertex set $\mathbb{P}(V)$, and where $(\bar{\mathbf{u}}, \bar{\mathbf{v}}) \in \mathbb{P}(V)^2$ is an edge in G_ϕ if and only if $\phi_{\mathbf{u},\mathbf{v}} = 0$. This graph is an invariant of alternating trilinear forms, introduced by Hora and Pudlák to classify all the trilinear forms over $\mathbb{F}_2$ in dimensions 8 and 9 [14,15].[1] Hora and Pudlák observed that

$$\deg(\bar{\mathbf{v}}) = \frac{q^{n-\mathrm{rank}(\mathbf{v})} - q}{q-1},$$

because $\bar{\mathbf{u}}$ is a neighbour of $\bar{\mathbf{v}}$ precisely if $\bar{\mathbf{u}} \neq \bar{\mathbf{v}}$ and $\mathbf{u} \in \mathrm{Rad}_\phi(\mathbf{v})$.

In the remainder of this section, we explore some statistics of the graphs of uniformly random trilinear forms. We compute the average number of points of each rank, and the number of edges between them.

Theorem 1. *Let $n \in \mathbb{N}$ and let $n-d$, $n-d_1$, and $n-d_2$ be non-negative even numbers less than n. Then, as q goes to infinity, the average number of projective points $\bar{\mathbf{u}} \in \mathbb{P}(\mathbb{F}_q^n)$ with $\mathrm{rank}(\bar{\mathbf{u}}) = n-d$ of a uniformly randomly trilinear form $\phi \in ATF(\mathbb{F}_q^n)$ tends to*

$$q^{(-d^2+3d)/2+n-2},$$

and the average number of ordered edges $(\bar{\mathbf{u}}_1, \bar{\mathbf{u}}_2)$ in the graph G_ϕ with $\bar{\mathbf{u}}_1$ of rank $n-d_1$ and $\bar{\mathbf{u}}_2$ of rank $n-d_2$ tends to

$$q^{\frac{-d_1^2-d_2^2+5(d_1+d_2)}{2}+n-6}.$$

[1] For dimension 8 over $\mathbb{F}_2$, the graph invariant distinguishes all the alternating trilinear forms up to Isomorphism, in dimension 9 more invariants are needed.

Examples. We apply the theorem to dimensions $9, 10$, and 11, because those are the parameters proposed by [20].

- In dimension $n = 9$, the graph of a random form ϕ has on average close to q^2 points of rank 4, q^7 points of rank 6, the average number of $(4,4)$-edges and $(4,6)$-edges tends to q^3 and q^6 respectively. This means that rank-4 points have on average q rank-4 neighbours and q^4 rank-6 neighbours, and each rank 6 point has on average $1/q$ neighbours of rank 4.
- In dimension $n = 10$, the average number of rank-4 points tends to $1/q$. We exploit this in Section 6 by giving a very efficient key-recovery attack that works for a $1/q$ fraction of all keys. The average number of rank-6 points is q^6.
- In dimension $n = 11$, the average number of rank-6 and rank-8 points tends to q^4 and q^9 respectively, each rank-6 point has on average close to q rank-6 neighbours and q^4 rank-8 neighbours. Rank-8 points have on average $1/q$ rank-6 neighbours.

Note that the first part of the statement agrees with the experimental observations in [20](Table 3), where the authors observed that for randomly chosen ϕ, there are close to $q^{n-1}, q^{n-3}, q^{n-6}$, and q^{n-10} vectors of rank $n-3, n-4, n-5$, and $n-6$ respectively. Finding a proof for these rank statistics was left as an open problem. Before we prove the theorem, we first approximate the probability that a random-skew symmetric matrix has a certain rank.

Lemma 1. *Let n, d be integers, such that $0 \leq n-d \leq n$ and $n-d$ is even, then*

$$\Pr_{M \leftarrow \mathcal{S}(q,n)}[M \text{ has rank } n-d] \sim q^{(-d^2+d)/2} \text{ as } q \to \infty$$

Proof. We first prove the case $d = 0$ and n even. Let M be a random skew-symmetric n-by-n matrix. Then $Me_1 = 0$ if the first column of M is zero, which happens with probability q^{1-n} (all the entries are uniformly random, except the first one which is zero). By symmetry, it follows that $\Pr[M\mathbf{v} = 0] = q^{1-n}$ for any non-zero vector $\mathbf{v} \in \mathbb{F}_q^n$, and by the linearity of expectation, the average number of non-zero vectors in the kernel of M is $(q^n - 1)q^{1-n} < q$.

Since skew-symmetric matrices have even rank, each singular skew-symmetric matrix has at least $q^2 - 1$ non-zero vectors in the kernel, so we have

$$(q^2 - 1)\Pr[M \text{ is singular }] < q\,,$$

which implies that M is non-singular with probability at least $1 - q/(q^2 - 1)$ which tends to 1 as q goes to infinity.

Now we move on to the case $d > 1$. The random matrix M has rank $n - d$ if it has a kernel of dimension d. The kernel of M is equal to $\langle e_1, \ldots, e_d \rangle$ if its first

d columns are zero, and if the remaining $(n-d)$-by-$(n-d)$ submatrix M' is non-singular, which happens with probability

$$\underbrace{q^{1-n}}_{\text{column 1}} \cdots \underbrace{q^{d-n}}_{\text{column } d} \Pr[M' \text{ non-singular}] = q^{-nd+(d^2+d)/2} \Pr[M' \text{ non-singular}],$$

where we know from the first part of the proof that the probability that M' is non-singular tends to 1 as q goes to infinity. By symmetry, this is not only true for $\langle e_1, \ldots, e_d \rangle$, but also for all the other spaces of dimension d. Therefore the total probability that M has rank $n-d$ is

$$\underbrace{\prod_{i=0}^{d-1} \frac{q^n - q^i}{q^d - q^i}}_{\#\text{ of } d\text{-spaces}} q^{-nd+(d^2+d)/2} \Pr[M' \text{ non-singular}] \sim q^{(-d^2+d)/2} \text{ as } q \to \infty. \quad \square$$

Theorem 1. For the first part of the theorem, it suffices to compute the probability that an arbitrary $\mathbf{v} \in \mathbb{P}(\mathbb{F}_q^n)$ has rank $r = n-d$, so let $\mathbf{v} \neq 0$ be an arbitrary non-zero vector in $\mathbb{F}_q^n$.

Extend $\mathbf{v}$ to a basis $\mathbf{v} = \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ of $\mathbb{F}_q^n$. Then with respect to this basis, the bilinear form $\phi_{\mathbf{v}}$ has a matrix representation $\phi_{\mathbf{v}}(\sum_i y_i \mathbf{v}_i, \sum_i z_i \mathbf{v}_i) = \mathbf{y}^t M \mathbf{z}$, with

$$M = \begin{pmatrix} 0 & 0_{1\times(n-1)} \\ 0_{(n-1)\times 1} & M' \end{pmatrix},$$

where M' is a uniformly random $(n-1)$-by-$(n-1)$ skew-symmetric matrix. Therefore, it follows from Lemma 1 that if $n-d$ is even, the probability that $\phi_{\mathbf{v}}$ has rank $n-d$ tends to

$$q^{(-(d-1)^2+(d-1))/2} = q^{(-d^2+3d-2)/2}.$$

The number of projective points in $\mathbb{P}(\mathbb{F}_q^n)$ tends to q^{n-1}, so by the linearity of expectation, the average number of points of rank $n-d$ tends to

$$q^{(-d^2+3d)/2+n-2}.$$

To prove the second part of the theorem, we similarly compute the probability that an arbitrary pair of distinct projective points $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{P}(\mathbb{F}_q^n)$ is an edge in G_ϕ, with $\mathbf{v}_1$ of rank $n-d_1$ and $\mathbf{v}_2$ of rank $n-d_2$.

Extend $\mathbf{v}_1, \mathbf{v}_2$ to a basis $\mathbf{v}_1, \ldots, \mathbf{v}_n$ of $\mathbb{F}_q^n$. With respect to this basis, the bilinear forms $T_{\mathbf{v}_1}$ and $T_{\mathbf{v}_2}$ have matrix representations

$$M_{\mathbf{v}_1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \mathbf{a}^t \\ 0 & -\mathbf{a} & M_1' \end{pmatrix} \text{ and } M_{\mathbf{v}_2} = \begin{pmatrix} 0 & 0 & -\mathbf{a}^t \\ 0 & 0 & 0 \\ \mathbf{a} & 0 & M_2' \end{pmatrix},$$

where $\mathbf{a} \in \mathbb{F}_q^{n-2}$ and $M'_{\mathbf{v}_1}, M'_{\mathbf{v}_2} \in \mathbb{F}_q^{(n-2)\times(n-2)}$ are uniformly random. Now $([\mathbf{v}_1], [\mathbf{v}_2])$ is an edge in the graph G_T if and only if $\mathbf{a} = 0$, which happens with probability q^{-n+2}. And if this is the case, then $[\mathbf{v}_1]$ and $[\mathbf{v}_2]$ have ranks r_1 and r_2 if $M'_{\mathbf{v}_1}$ and $M'_{\mathbf{v}_2}$ have ranks r_1 and r_2 respectively. Lemma 1 says that this happens with probabilities that tend to

$$q^{\frac{(d_1-2)^2+(d_1-2)}{2}} \text{ and } q^{\frac{(d_2-2)^2+(d_2-2)}{2}}$$

respectively. Since the number of ordered pairs of projective points tends to q^{2n-n}, it follows from the linearity of expectation that the average number of $(n-d_1, n-d_2)$-edges in G_ϕ tends to

$$q^{2n-2}q^{-n-2}q^{\frac{(d_1-2)^2+(d_1-2)}{2}}q^{\frac{(d_2-2)^2+(d_2-2)}{2}} = q^{n-6+\frac{-d_1^2-d_2^2+5(d_1+d_2)}{2}}. \qquad \square$$

4 Solving ATFE with auxiliary information via Gröbner bases.

This section describes the Gröbner basis-based approach to solving the ATFE problem with auxiliary information, which we will use as a subroutine for our algorithms to solve ATFE without auxiliary information. We also give some improvements that make the subroutine run faster in practice.

The algorithm of [6] Bouillaguet, Faugère, Fouque, and Perret suggest solving the ATFE problem, using a Gröbner basis approach [6]. This works as follows: let S be an n-by-n matrix whose entries $\{s_{ij}\}$ we consider to be formal variables and use a system-solving algorithm to find the solutions of the system

$$\phi_2(x, y, z) = \phi_1(Sx, Sy, Sz)\,.$$

This is a system of $\binom{n}{3}$ cubic equations in the n^2 variables $\{s_{ij}\}$, and the solutions precisely correspond to the isomorphisms from ϕ_1 to ϕ_2. We can also consider a second matrix T (which represents the inverse of S) with entries $\{t_{ij}\}$, and then solve the system

$$\begin{cases} ST = TS = \mathbf{1}_n \\ \phi_2(x, y, z) = \phi_1(Sx, Sy, Sz) \\ \phi_2(x, y, Tz) = \phi_1(Sx, Sy, z) \\ \phi_2(x, Ty, Tz) = \phi_1(Sx, y, z) \\ \phi_2(Tx, Ty, Tz) = \phi_1(x, y, z) \end{cases}.$$

This is a system of $2n^2 + 4\binom{n}{3}$ equations in $2n^2$ variables. It turns out that solving the second system is more efficient because we now have many quadratic equations, which are generally easier to solve than cubic equations. Nevertheless, it seems that solving the systems of equations is still exponentially hard. An instance of the problem with $n = 7, q = 16$ was solved by [6] in five hours with 3 GB of RAM, but the same approach failed for $n = 8$ after running out of

memory (74 GB was available). However, Bouillaguet *et al.* observed that the problem becomes much easier when some auxiliary information in the form of $\mathbf{u}, \mathbf{v} \in \mathbb{F}_q^n$ such that $S\mathbf{v} = \mathbf{u}$ is available. In this case, after adding the n linear equations $S\mathbf{v} = \mathbf{u}$ to the system, the system can be solved at degree 2 with time complexity $O(n^6)$. This makes it possible to solve an instance with $n = 16, q = 2$ in only 90 seconds [6].

Improvements and experiments. We observe that we can use slightly less auxiliary information and still have an efficient algorithm: we can use a pair of projective points $\bar{\mathbf{u}}, \bar{\mathbf{v}} \in \mathbb{P}(\mathbb{F}_q^n)$ such that $S\bar{\mathbf{u}} = \bar{\mathbf{v}}$, rather than affine points $\mathbf{u}, \mathbf{v}$ such that $S\mathbf{u} = \mathbf{v}$ exactly. This means that the system of equations has just one fewer linear equation, which does not seem to affect the running time of the Gröbner basis algorithm too much. This is advantageous because finding projective points $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$ such that $S\bar{\mathbf{u}} = \bar{\mathbf{v}}$, is easier than finding affine points $\mathbf{u}, \mathbf{v}$ such that $S\mathbf{u} = \mathbf{v}$.

The algorithm of [20] and our algorithms of Section 5.3 solve instances of the ATFE problem with auxiliary information $\bar{\mathbf{v}} = S\bar{\mathbf{u}}$, where $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$ are points of low rank for ϕ_1 and ϕ_2 respectively. We can exploit this to speed up the system-solving approach because we know that $S\,\mathrm{Rad}_{\phi_2}(\mathbf{v}) = \mathrm{Rad}_{\phi_1}(\mathbf{u})$ and $\mathrm{Rad}_{\phi_2}(\mathbf{v}) = T\,\mathrm{Rad}_{\phi_1}(\mathbf{u})$. This gives $2R(n - R - 1)$ additional linear equations on the variables $\{s_{ij}\}$ and $\{t_{ij}\}$, which can be used to eliminate some variables to make the system solving approach more efficient in practice.

We do some experiments using the Gröbner basis implementation of the Giac library which is accessible through Sage [17,22] on the authors' laptop (intel i9-10885H CPU and 64 GB of RAM). We solve some instances of the ATFE problem with auxiliary information that we will need in Section 5.3. Unlike Bouillaguet *et al.* , who did experiments for trilinear forms that were not necessarily alternating and with auxiliary points that did not have low rank, we notice that the Gröbner basis algorithm does not always terminate at degree 2. Nevertheless, the algorithm is efficient enough to run in practice in a reasonable amount of time. The results are given in Table 2. We notice that the Gröbner-basis solving algorithm always finds three solutions, because if (S, T) is a solution, then $(\alpha S, \alpha^2 T)$ and $(\alpha^2 S, \alpha T)$ are also solutions, where $\alpha \in \overline{\mathbb{F}}_q$ is a third root of unity.

Table 2. Solving several ATFE problems with auxiliary information with SAGE.

(n, q)	Rank of $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$	Number of variables	Time (seconds)	Memory (MB)
$(9, 524287)$	6	122	270	4312
	4	114	5	305
$(10, 131071)$	6	146	1000	8971
	4	142	4800	18486
$(11, 65521)$	6	174	200	1097

5 Algorithms for the alternating trilinear form equivalence problem.

This section describes some heuristic algorithms for the ATFE problem. We first revisit the algorithms of [6,20], before introducing new algorithms.

5.1 The algorithms of [6,20]

$O(q^n n^6)$ algorithm. It follows from the Algorithm in the previous section that to solve the ATFE problem it suffices to find a good pair of vectors $\mathbf{u}, \mathbf{v}$ such that $S\mathbf{v} = \mathbf{u}$. Just taking a pair $(\mathbf{e}_1, \mathbf{u})$ where $\mathbf{u} \in \mathbb{F}_q^n$ is uniformly random has a success probability of $|\mathrm{Aut}(\phi)|q^{-n}$. Randomly chosen alternating trilinear forms in dimension $n \geq 10$ seem to have no non-trivial automorphisms, so for random trilinear forms, the algorithm of [6] runs in time $O(q^n n^6)$.

$O(q^{2n/3}\mathrm{poly}(n))$ algorithm. The authors of [20] improve on this approach, using the fact that if $\mathbf{u}$ has rank r for ϕ_2, then $S\mathbf{u}$ must also have rank r for ϕ_1. Their algorithm is parametrized by a rank R. Let N_R be the number of vectors in $\mathbb{F}_q^n$ with rank R for ϕ_1. Then in the first phase the algorithm computes two lists L_1, L_2 of $O(N_R^{1/2})$ vectors of rank R for ϕ_1 and ϕ_2 respectively. This is done by brute force: repeatedly pick $\mathbf{u} \in \mathbb{F}_q^n$ at random, compute the rank of $\phi_{1\mathbf{u}}$ and keep the vector if the rank is equal to R. This takes on average q^n/N_R attempts per vector, so it makes for a total cost of $O(q^n N_R^{-1/2}\mathrm{poly}(n))$. Then with high probability, there is a vector $\mathbf{u} \in L_1$ and $\mathbf{v} \in L_2$ such that $S\mathbf{u} = \mathbf{v}$. We do not know which pair is the good one, so in the second phase, we run the system-solving algorithm of Section 4 for every pair $(\mathbf{u}, \mathbf{v}) \in L_1 \times L_2$, for a total cost of $O(N_R n^6)$. The complexity of the algorithm is optimal if the rank R is chosen such that $N_R \approx q^{3n/2}$. If such an R exists, we get an algorithm that runs in time $O(q^{2n/3}\,\mathrm{poly}(n))$.

5.2 A general MinRank-based algorithm.

Using MinRank solving algorithms. We generalize the algorithm of [20] to allow for more efficient algorithms to build the lists L_1 and L_2. From the trilinearity of ϕ, we have that $\phi_{\mathbf{v}} = \sum_i v_i \phi_{\mathbf{e}_i}$, where v_i is the i-th coordinate of $\mathbf{v}$ with respect to the standard basis $\mathbf{e}_i, \ldots, \mathbf{e}_n$. Therefore, as observed by [20], finding the coefficients of a low-rank vector $\mathbf{v}$ for ϕ is equivalent to finding a low-rank linear combination of the alternating bilinear forms $\phi_{\mathbf{e}_1}, \ldots \phi_{\mathbf{e}_n}$, which is an instance of the MinRank problem of Section 2. The algorithm uses a target rank R, and samples $O(\sqrt{N_R})$ (projective) solutions to the MinRank instances corresponding to ϕ_1 and ϕ_2. Then we try all the $O(N_R)$ pairs of points $(\bar{\mathbf{u}}, \bar{\mathbf{v}}) \in L_1 \times L_2$. This makes for a complexity of

$$O(\sqrt{N_R}\,\mathsf{MR}_{n,r} + N_R n^6)\,,$$

where N_R is now the number of *projective* points of rank R, and where $\mathsf{MR}_{n,R}$ denotes the complexity of sampling a solution to the MinRank instance derived

from an alternating trilinear form ϕ in dimension n, and with target rank R. Unfortunately, it seems cumbersome to get good estimates of $\mathsf{MR}_{n,R}$. This is because the matrices in our MinRank problem are structured: they represent the bilinear forms $\phi_{\mathbf{e}_i}$, so the matrices are antisymmetric, have zeroes on the i-th row and column, and $(\phi_{\mathbf{e}_i})_{j,k} = -(\phi_{\mathbf{e}_j})_{i,k} = (\phi_{\mathbf{e}_k})_{i,j}$. This seems to adversely affect the performance of the MinRank solving algorithm of Bardet et al. [1] because there are non-trivial linear dependencies between the equations in the support minors modeling. It would be interesting to rigorously investigate how this structure affects the performance of existing MinRank-solving algorithms, and perhaps design more efficient MinRank-solving algorithms that can take advantage of this structure, especially for large n. We do not estimate the running time of this algorithm and we move on to the more efficient approach of the next subsection where rather than solving MinRank instances to populate the lists L_1 and L_2, we use an approach that is more efficient for small n which exploits walking in the G_ϕ graph.

5.3 Graph-walking algorithms for small n.

In this section, we exploit the fact that once a low-rank point $\bar{\mathbf{v}} \in \phi$ is found, we can look for its low-rank neighbours in the graph G_ϕ to find additional low-rank vectors more efficiently.

Odd dimensions. In odd dimensions, according to Theorem 1, there are approximately q^{n-2} points of rank $n-3$, which makes a $1/q$ fraction of all points. So, to sample a point of rank $n-3$ by brute force takes $O(q\,\mathrm{poly}(n))$ work. However, once a single point $\bar{\mathbf{v}}$ of rank $n-3$ is found, we can sample additional rank-$(n-3)$ vectors with only $O(\mathrm{poly}(n))$ effort, by just sampling one of the neighbours of $\bar{\mathbf{v}}$.

Lemma 2. *Let n be odd, and let $\bar{\mathbf{v}}$ be a projective point with rank at most $n-3$ for an alternating trilinear form ϕ. Then the neighbours of $\bar{\mathbf{v}}$ in the graph G_ϕ also have rank at most $n-3$.*

Proof. Suppose $\bar{\mathbf{v}}$ is a neighbour of $\bar{\mathbf{u}}$, then we have that $\langle \bar{\mathbf{u}}, \bar{\mathbf{v}} \rangle \in \mathrm{Rad}(\bar{\mathbf{v}})$, so the rank of $\bar{\mathbf{v}}$ is $n - \dim(\mathrm{Rad}(\bar{\mathbf{v}})) < n-2$, but the rank has to be even, so $\bar{\mathbf{v}}$ must have rank at most $n-3$. □

Is it useful to generate many points of rank $n-3$? We could use the algorithm from the previous subsection at rank $R = n-3$, but that would get a complexity q^{n-2}, which does not improve on the brute-force algorithm of [20]. However, rank-$(n-3)$ points are quite likely to have rank-$(n-5)$ neighbours. Theorem 1 says that the number of $(n-3, n-5)$-edges in G_ϕ is approximately q^{n-3}. Recall that there are q^{n-2} rank $(n-3)$ points, so we expect that a $1/q$-fraction of rank $n-3$ points have a rank-$(n-5)$ neighbour.

This suggests the following approach for sampling rank $n-5$ points: starting from a point of rank $n-3$, do a random walk in G_ϕ, and at each point $\bar{\mathbf{v}}$ check if $\bar{\mathbf{v}}$ has

a neighbour of rank $n-5$. We expect this to succeed after $O(q)$ steps. Checking if $\bar{\mathbf{v}}$ has a rank $n-5$ neighbour is efficient, because the neighbours of $\bar{\mathbf{v}}$ form a space of dimension 3, so we only have to solve a MinRank instance with 3 matrices of size n-by-n and target rank $n-5$. We can do this with the support-minors algorithm of [1] with a complexity of $O(k^3 r\binom{n}{r}^2) = O(n\binom{n}{n-5}^2) = O(n^{11})$. The complexity $O(n^{11})$ might seem impractical, but the bottleneck of the algorithm is doing linear algebra on a sparse matrix with $3\binom{n}{5}$ columns, which for $n = 9$ and $n = 11$ is only 378 and 1386 columns respectively, which is still very practical.

If we run the algorithm from the previous section at rank $R = n - 5$ with the graph-walking approach we get a complexity of

$$O(q^{(n-5)/2} n^{11} + q^{n-7} n^6).$$

Which for small n is better than the algorithm of [20]. For large enough q, the running time of the algorithm is dominated by the second phase, which runs Gröbner basis algorithms for each of the approximately $N_{n-5} = q^{n-7}$ pairs $(\bar{\mathbf{u}}, \bar{\mathbf{v}}) \in L_1 \times L_2$. Two of the three parameter sets proposed by [20] have odd n, these are $(n, q) = (9, 524287)$ and $(n, q) = (11, 65521)$. For these parameters the bottleneck of the algorithm is running $524287^2 \approx 2^{38}$ and $65521^4 \approx 2^{64}$ executions of the Gröbner basis algorithm respectively. Given that a Sage implementation of the Gröbner basis algorithm takes approximately 5 seconds and 200 seconds for these parameter sets (see Table 2), we can conclude that these parameters fall short of their target security level of 128 bits of security. Nevertheless, a practical break still seems out of reach.

Even dimension. We now adapt the attack to the case of an even dimension. Note that in this case, all points have rank at most $n - 2$, since the rank has to be even, and less than n. Moreover, according to Theorem 1, the number of $(n-2, n-4)$ edges and the number of rank-$(n-2)$ points are both q^{n-1}, so we expect most rank-$(n-2)$ points to have one rank-$(n-4)$ neighbour. Finding a rank-$(n-4)$ neighbour of a rank-$(n-2)$ point $\bar{\mathbf{u}}$ comes down to solving a MinRank problem with only 2 n-by-n matrices M_1, M_2, corresponding to $\phi_{\mathbf{u}}, \phi_{\mathbf{v}}$, where $\mathbf{u}, \mathbf{v}$ is a basis for $\mathrm{Rad}_\phi(\mathbf{u})$. This problem can be solved by computing and factoring the gcd of a few determinants of $(n-3)$-by-$(n-3)$ minors of $M_1 + \lambda M_2$. This approach allows us to sample points of rank $n-4$ with $O(n^3)$ field operations, resulting in an algorithm for the ATFE problem with complexity

$$O(\sqrt{N_{n-4}}\, n^3 + N_{n-4}\, n^6) = O(q^{(n-4)/2} n^3 + q^{n-4}\, n^6)\,.$$

For $n = 10$, this becomes a $O(q^6)$ algorithm, only slightly better than the algorithm of [20] which has a complexity of $O(q^7)$. We believe it is better to use rank $R = n - 6$ for larger n, but we leave an analysis of the complexity for future work since we are mostly interested in dimensions $n = 9, 10$ and 11.

5.4 A (sketch of an) algorithm using graph-neighbourhood invariants.

We would like to have more invariants to distinguish points in alternating trilinear forms, by which we mean functions

$$F : \mathrm{ATF}(\mathbb{F}_q^n) \times \mathbb{P}(\mathbb{F}_q^n) \to X : (\phi, \bar{\mathbf{v}}) \mapsto F(\phi, \bar{\mathbf{v}}) \,,$$

such that $F(\phi, \bar{\mathbf{v}}) = F(\phi \circ S, S^{-1}\bar{\mathbf{v}})$ for all $S \in GL(\mathbb{F}_q^n)$. We already heavily used one of these invariants, namely $\mathrm{rank}_\phi(\bar{\mathbf{v}})$, but if we had a more powerful invariant we could speed up the second phase of our ATFE algorithm. Instead of running the Gröbner basis algorithm for all pairs $(\bar{\mathbf{u}}, \bar{\mathbf{v}}) \in L_1 \times L_2$, we only need to consider pairs $(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ such that $F(\phi_1, \bar{\mathbf{u}}) = F(\phi_2, \bar{\mathbf{v}})$. If the invariant is sufficiently powerful such that there are no false positives, then we would only have to do a single Gröbner basis computation. This would reduce the complexity of the attack to

$$O(\sqrt{N_R}\ (\mathsf{MR}_{n,R} + T_F) + n^6) \,,$$

where T_F is the time it takes to compute the invariant F. We have a few candidates for invariants, based on the graphs of alternating trilinear forms. We know that if $\phi_2 = \phi_1 \circ S$, then the graphs G_{ϕ_1} and G_{ϕ_2} are isomorphic, and we have an explicit isomorphism given by $\Psi_S : G_{\phi_2} \to G_{\phi_1} : \bar{\mathbf{v}} \mapsto S\bar{\mathbf{v}}$. Similarly, the restrictions of G_{ϕ_1} and G_{ϕ_1} to points of rank R are also isomorphic with the same isomorphism. Therefore, following the approach of Bouillaguet et al. [7] for the closely related "Isomorphism of Polynomials" problem, we can define a family of invariants $F_{k,r}(\phi, \bar{\mathbf{v}})$ that outputs a canonical representation of the radius-k-neighbourhood of $\bar{\mathbf{u}}$ in the restriction of G_ϕ to points of rank r.

In dimension 9, it turns out that with high probability the graph is vertex-transitive, which means the graph invariants do not have any distinguishing power. However, for larger n, preliminary experiments suggest that the graphs invariants can distinguish points quite well, even for small radii (e.g. 1 or 2), so we conjecture that this approach gives rise to an algorithm that runs in time $O(q^{n/2+c}\,\mathrm{poly}(n))$ for some constant c.

6 A class of weak keys for $n = 10$

Theorem 1 says that a random alternating trilinear form in dimension $n = 10$ has on average q^{-1} points of rank 4, which suggests that a random trilinear form has a unique rank-4 point with probability close to $1/q$. We can confirm this experimentally for small q. Moreover, if the rank-4 point exists, we can find it efficiently by solving a MinRank problem with 10 matrices of dimensions 10-by-10 and a target rank of 4. This gives an efficient 2-step attack on a $1/q$ fraction of all public keys: Firstly, try to find the unique rank-4 points $\bar{\mathbf{u}}, \bar{\mathbf{v}}$ in ϕ_1 and ϕ_2 respectively. Secondly, if the points exist, run the Gröbner basis algorithm from Section 4 to find the equivalence S such that $\phi_2 = \phi_1 \circ S$, using the auxiliary information that $\bar{\mathbf{u}} = S\bar{\mathbf{v}}$.

We observe experimentally that this works quite well in practice for the $(n, q) = (10, 131071)$ parameter set proposed in [20]. Our Sage script can find the rank-4 point in ϕ_1 and ϕ_2 in roughly 16 minutes each, by solving the associated MinRank problem. After finding the two rank-4 points (which can be done in parallel using two cores) we can solve for the equivalence $S \in GL(\mathbb{F}_q^{10})$ such that $\phi_2 = \phi_1 \circ S$ with the Gröbner basis approach from Section 4. This takes approximately 1 hour and 20 minutes (see Table 2). The total attack takes approximately one hour and 36 minutes on the author's laptop.[2]

7 The curious case of $n = 9$

We observe that for randomly chosen alternating trilinear forms in dimension 9, the restriction of G_ϕ to points of rank 4 has a lot of structure that we do not fully understand. In this section, we exploit this structure to formulate a heuristic algorithm for the ATFE problem in dimension 9. Our algorithm runs in time $O(q)$ and works well in practice. We can solve the ATFE problem for the $n = 9$ parameter set proposed by [20] in at most 4 hours on the author's laptop. We also observe that if the number of points of rank 4 is divisible by 2^r (large powers of two seem to occur quite frequently for reasons we do not understand), then the attack runs in time $O(q/\sqrt{2^r})$. This gives a family of weak keys for which the attack is faster.

7.1 Graph-neighbourhood invariants.

According to Theorem 1, a random alternating trilinear form $\phi \in \mathrm{ATF}(\mathbb{F}_q^9)$ has on average close to q^2 points of rank 4, and the graph G_ϕ, restricted to the rank-4 points has on average $q^3/2$ edges, meaning that the average point in the graph has degree q. We start our investigation by computing and plotting the rank-4 graphs for some 9-dimensional alternating trilinear forms: Figure 1 displays the rank-4 graphs of a selection of five typical alternating trilinear forms modulo $q = 5$. We see that the graphs are surprisingly nice:

Observation 1. *Let $\phi \in ATF(\mathbb{F}_q^9)$ be a uniformly random form, let $G_{4,\phi}$ be the restriction of G_ϕ to points of rank 4, and let $N_4 = |G_{4,\phi}|$ be the number of points of rank 4. Then with high probability $G_{4,\phi}$ has dihedral symmetry Dih_{N_4}. In particular, $G_{4,\phi}$ is vertex-transitive and has at least $2N_4$ automorphisms.*

Figure 2 shows some counterexamples, where the graphs are not regular or do not have the dihedral symmetry, but even these counterexamples are 'nice' in the

[2] To do the experiments, we deliberately generate weak keys with a point of rank 4, by first generating a random alternating trilinear form ϕ' for which $\mathrm{Rad}'_\phi(\mathbf{e}_1) = \langle \mathbf{e}_1, \ldots, \mathbf{e}_6 \rangle$, so that $\mathbf{e}_1$ has rank 4 and then composing ϕ with a random invertible map $T \in GL(\mathbb{F}_q^{10})$ to send the rank-4 point to some random position. Every form with a point of rank-4 is isomorphic to a form ϕ' where $\mathrm{Rad}_{\phi'}(\mathbf{e}_1) = \langle \mathbf{e}_1, \ldots, \mathbf{e}_6 \rangle$, so this method of generating instances does not introduce additional structure that could affect the hardness of finding the rank-4 point or the Gröbner basis step.

sense that they have a lot of automorphisms and that they are almost regular: there are at most two distinct degrees. For $q = 5$ the irregular graphs are still fairly common, making up about 20% of the randomly sampled forms, but for larger q the irregular graphs seem to become increasingly rare. We computed the rank-4 graph for 25 randomly chosen forms with $q = 31$, and they were all regular with dihedral symmetry Dih_{N_4}. This is unfortunate because it means the graph-based invariants from Section 5.4 are completely useless in dimension $n = 9$. Since the graph is vertex-transitive, the neighbourhoods of any two vertices look the same, so we can not use the graphs to distinguish good pairs $(\mathbf{u}, \mathbf{v})$ from bad pairs.

7.2 A mysterious function H

Since the graph-based invariants fail, we try to construct some other invariants. We start only from a form ϕ and a point $\bar{\mathbf{v}} \in \mathbb{P}(\mathbb{F}_q^9)$ of rank 4. What objects can we build from this information? We have $K = \mathrm{Rad}_\phi(\mathbf{v})$, which is a 5-dimensional space, so it is natural to look at $\phi|_K$, the restriction of ϕ to K. The isomorphism class of this 5-dimensional form is an invariant. Unfortunately, this invariant is not very powerful, because $\phi|_K$ is degenerate ($\phi(\mathbf{v}, \mathbf{x}, \mathbf{y}) = 0$ for all $\mathbf{x}, \mathbf{y} \in K$), and it turns out that up to isomorphism there are only two degenerate forms in dimension 5, namely 0 and $\mathbf{e}_1^* \wedge \mathbf{e}_2^* \wedge \mathbf{e}_3^*$, so we get at most one bit of information from the invariant. Even worse, $\phi|_K = 0$ is extremely unlikely, so we get no information from the isomorphism class of $\phi|_K$. However, $\mathrm{Rad}(\mathbf{e}_1^* \wedge \mathbf{e}_2^* \wedge \mathbf{e}_3^*) = \langle \mathbf{e}_4, \mathbf{e}_5 \rangle$ is two-dimensional, so we have identified a new space $R = \mathrm{Rad}(\phi|_K) \subset K$.

We can use this space R to speed up the system-solving algorithm from Section 4. We know that if $\phi_2 = \phi_1 \circ S$ and $\bar{\mathbf{u}} = S\bar{\mathbf{v}}$ a good pair of points, then we must have that $SR_{\phi_2,\mathbf{v}} = R_{\phi_1,\mathbf{u}}$ and $R_{\phi_2,\mathbf{v}} = TR_{\phi_1,\mathbf{u}}$, which gives us some additional linear constraints on the entries of S and T, meaning that we can eliminate some more variables before applying the system-solving algorithm.

Encouraged by this small victory, we keep looking for more canonical spaces that we can construct from a 9-dimensional form ϕ and a rank-4 point $\bar{\mathbf{u}}$. We have $q + 1$ points in R, namely $\bar{\mathbf{u}}$ and q new points. Let $\bar{\mathbf{v}} \in R$ be one of the new points. We can look at its radical $\mathrm{Rad}_\phi(\bar{\mathbf{v}})$, which has dimension at least 3, because it has to be odd, and it includes R. This space is not canonical, because it relies on the choice of $\bar{\mathbf{u}}$ in $R \setminus \{\bar{\mathbf{u}}\}$, but the sum of all these spaces is canonical. This suggests we define a new space

$$W = \sum_{\substack{\bar{\mathbf{v}} \in R \\ \bar{\mathbf{v}} \neq \bar{\mathbf{u}}}} \mathrm{Rad}_\phi(\bar{\mathbf{v}})\,.$$

According to our experiments, this space is four-dimensional with high probability, and not included in K. This space W is covariant, so it also gives us some extra linear constraints on S and T that help speed up the system-solving algorithm. However, we observe something remarkable:

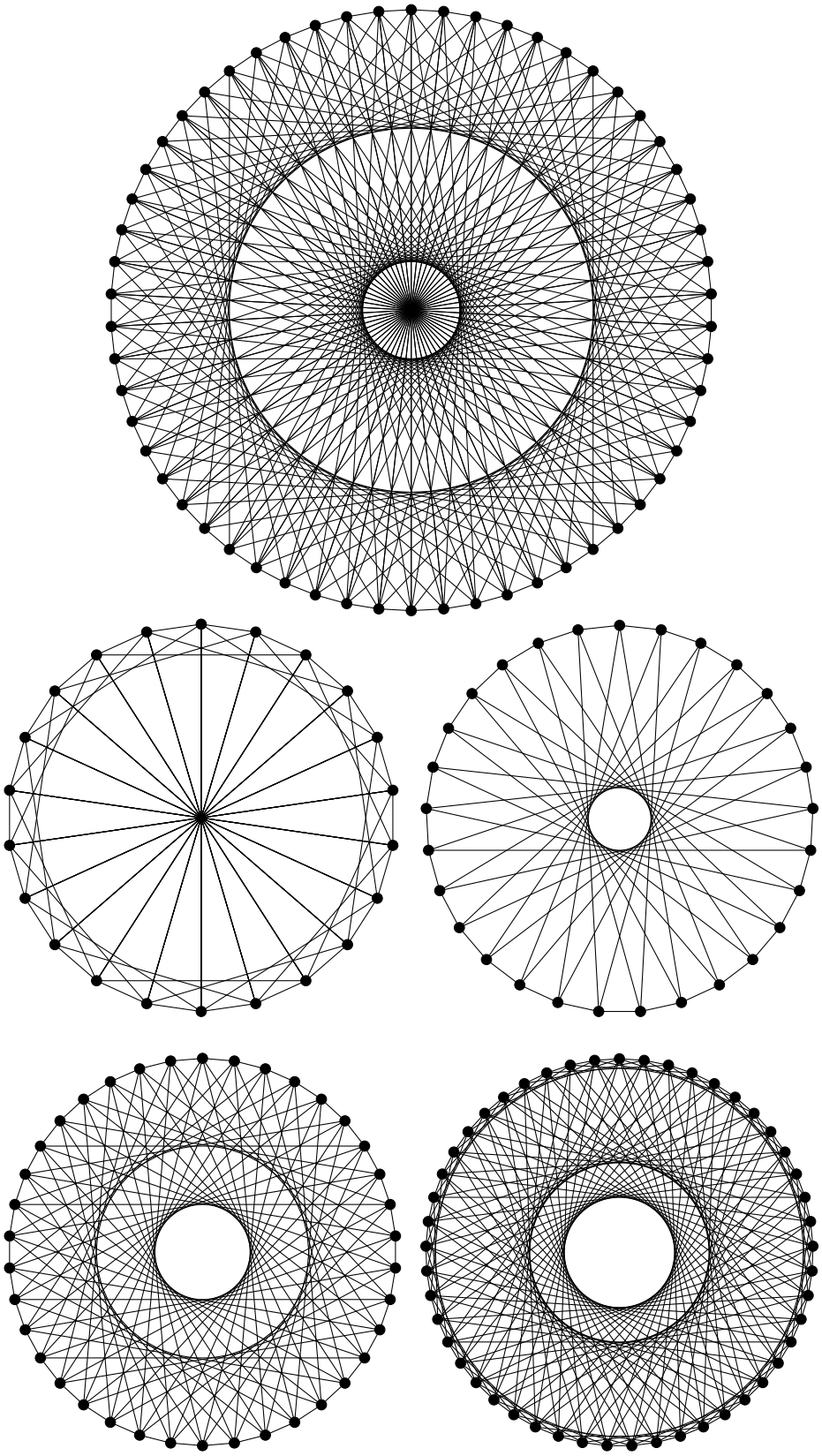

Fig. 1. Five typical rank-4 subgraphs of G_ϕ for randomly chosen alternating trilinear forms $\phi \in \mathrm{ATF}(\mathbb{F}_5^9)$. The graphs are regular and have Dihedral symmetry.

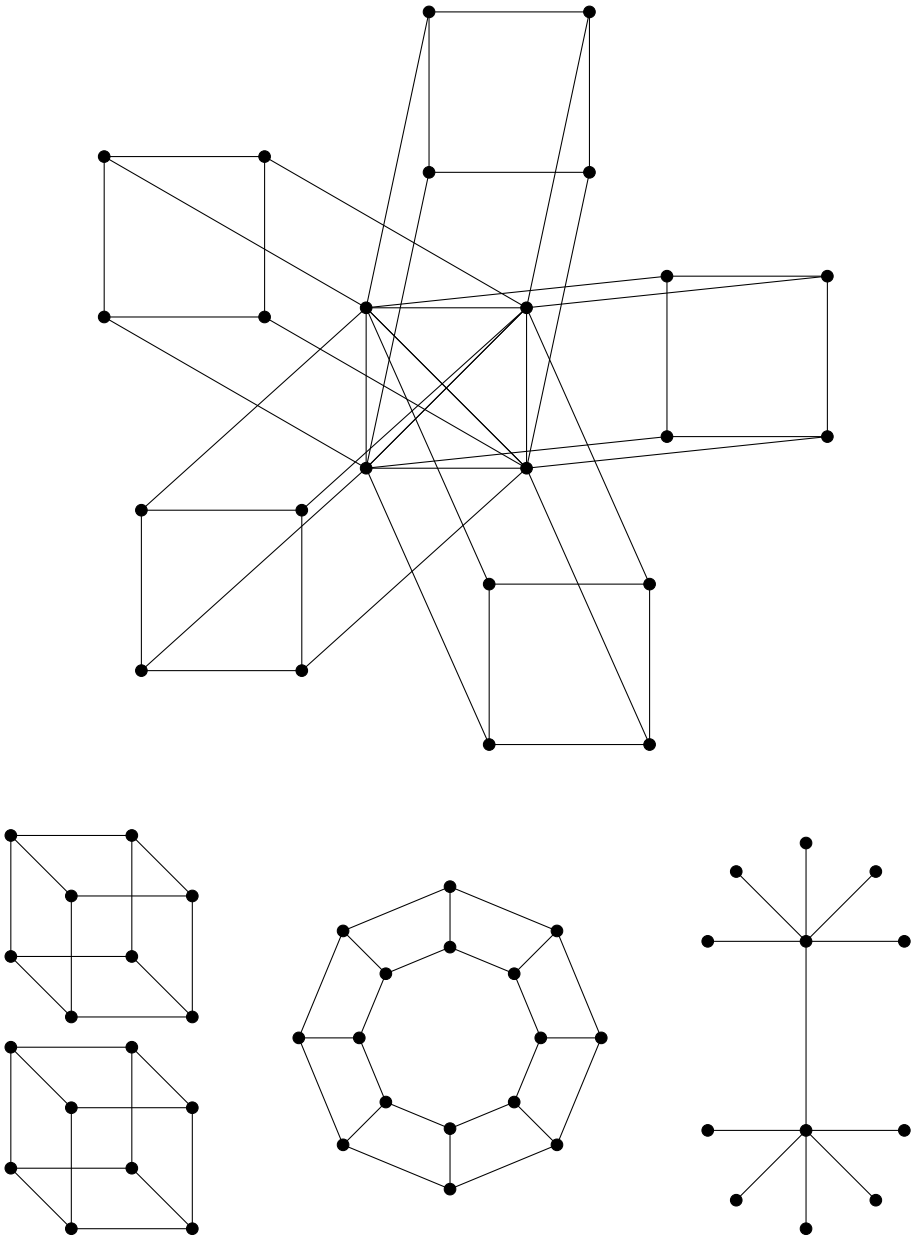

Fig. 2. A selection of four 'atypical' rank-4 graphs for alternating trilinear forms on $\mathbb{F}_5^9$. The top graph has graph automorphism group $Dih_4 \times S_5$, and the bottom graphs have automorphism groups (from left to right) $\mathbb{Z}_2 \times (\mathbb{Z}_2 \times S_4)^2$, $\mathbb{Z}_2 \times Dih_8$, and $\mathbb{Z}_2 \times S_5^2$.

Observation 2. *For random $\phi \in ATF(\mathbb{F}_q^9)$ and $\bar{\mathbf{u}} \in \mathbb{P}(\mathbb{F}_q^9)$ of rank 4, with high probability, we have that the space W contains exactly two points of rank 4, $\bar{\mathbf{u}}$ and some other point that we call $H_\phi(\bar{\mathbf{u}})$. With high probability this defines a function H_ϕ, mapping the set of rank-4 points of ϕ to itself.*

For small q, it happens sometimes that the function H_ϕ is not well-defined, because W could contain more than 2 rank-4 points, or only the one point $\bar{\mathbf{u}}$. We observe that as q grows, the function H_ϕ seems well defined at every point with increasingly high probability. To investigate this mysterious function H, we calculate and plot the directed graph of H_ϕ whose nodes are the rank-4 points of ϕ, and where there is an edge from $\bar{\mathbf{v}}$ to $\bar{\mathbf{u}}$ if $H_\phi(\bar{\mathbf{v}}) = \bar{\mathbf{u}}$. Some results can be seen in Figure 3. Even though the definition of H might seem arbitrary, we can see that the graphs have remarkable structure! We observe that if the number of rank four points is $2^d k$, with k odd, then the graph of H_ϕ is a collection of 2-volcano graphs with height d. That is, the nodes of each connected component can be partitioned in $d+1$ levels $V_0, \ldots, V_d$, where V_0 is a regular graph of degree $0, 1$ or 2, where $H(V_{i+1}) = V_i$ for $0 \leq i < d$, where each node in V_0 has exactly one incoming edge from V_1, and where each node in V_i has exactly two incoming edges from V_{i+1} for $1 \leq i < d$. The subgraph V_0 is called the *crater*, and V_d is called the *floor*. These ℓ-volcano graphs also appear in the study of isogenies between elliptic curves over finite fields [19]. The graphs on the left side of Figure 3 are drawn in a way to emphasize the 2-volcano structure. The same graphs are drawn on the right-hand side where the points are arranged in the circle according to the dihedral symmetry of G_ϕ. More precisely, we pick an elementary rotation r in $D_{N_4} \subset \mathrm{Aut}(G_\phi)$, and an arbitrary base point $\bar{\mathbf{u}}_0$. Then, since r acts transitively, we can label the rank-4 points as $\bar{\mathbf{u}}_i = r^i(\bar{\mathbf{u}}_0)$ for i going from 0 to $N_4 - 1$. We plot the point $\bar{\mathbf{u}}_i$ on the i-th location along a circle. From this plot, it is clear that $H_\phi(\bar{\mathbf{u}}_i) = \bar{\mathbf{u}}_{-2i+a \mod N_4}$ for some value of a. Note that this explains the 2-volcano structure. The value of a is not meaningful since it depends on the elementary rotation r and the base point $\bar{\mathbf{u}}_0$ that we chose to label the points. In the first graph of Figure 3, the function H_ϕ is not defined at the three points in V_0, since for these points there is no second rank-4 point in W. If we define $H_\phi(\bar{\mathbf{u}}) = \bar{\mathbf{u}}$ for these three points, then the equality $H_\phi(\bar{\mathbf{u}}_i) = \bar{\mathbf{u}}_{-2i+a \mod N_4}$ still holds.

7.3 Turning H into an invariant

Since the function H does not depend on a choice of a coordinate system, we have that H is covariant, i.e.

$$H_{\phi \circ S}(S^{-1}\bar{\mathbf{v}}) = S^{-1} H_\phi(\bar{\mathbf{v}})$$

for any $S \in GL(\mathbb{F}_q^n)$. We can turn this into an invariant by iterating the function and using projective frames. Starting from $\bar{\mathbf{v}}$, we build a canonical projective frame by iterating the function H_ϕ on $\bar{\mathbf{v}}$. This gives a sequence $\bar{\mathbf{v}}, H_\phi(\bar{\mathbf{v}}), H_\phi^2(\bar{\mathbf{v}}), \ldots$ from which we can drop an element if it is not independent of its predecessors. We continue this procedure until we have full projective

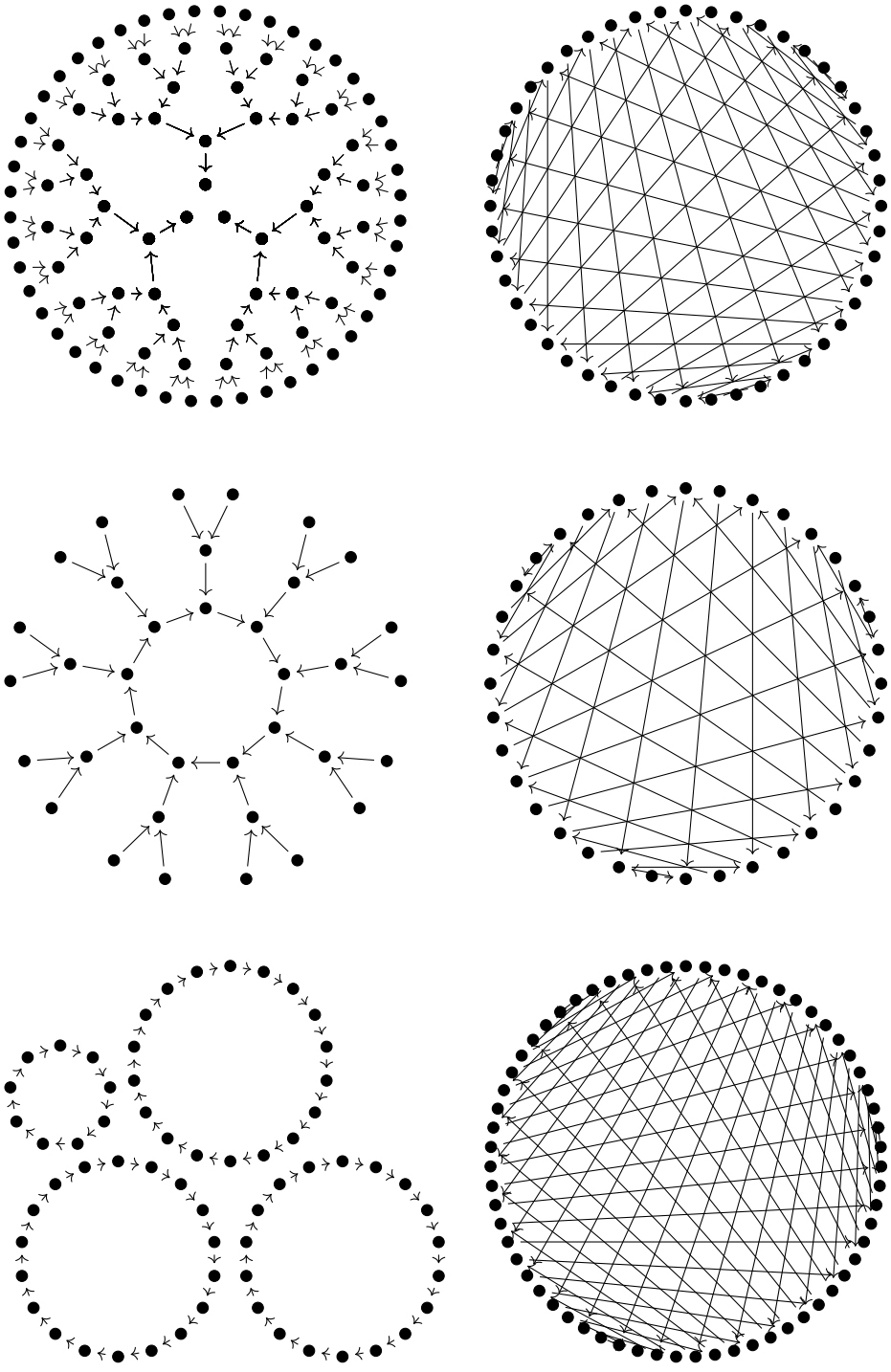

Fig. 3. Directed graphs corresponding to the H-function for three alternating trilinear forms. Each graph is drawn twice: once on the left in a way to make the 2-volcano structure of the graph clear, and once on the right where we arrange the rank-4 points in a circle according to the symmetry of G_ϕ. The three forms have 48, 36, and 63 points of rank 4 respectively. For the form on the first row, H is not defined at three points.

frame $\bar{\mathbf{v}}_1, \ldots, \bar{\mathbf{v}}_{10}$. It could happen that we never get a full projective frame, e.g. if the sequence is periodic with a period less than 10, but this does not seem to happen often for large enough q.

Now, to create an invariant, we sample one additional element $\bar{\mathbf{v}}_{11}$ from the sequence $\{H_\phi^i(\bar{\mathbf{v}})\}$ and we write it in homogeneous coordinates with respect to our canonical frame. These coordinates are an invariant, because if we were to compute the invariant for $\phi \circ S, S^{-1}\bar{\mathbf{u}}$, then we get the sequence $H_{\phi\circ S}(S^{-1}\bar{\mathbf{u}})$, which is equal to $S^{-1}H_\phi(\bar{\mathbf{u}})$. This means that the canonical projective frame is just $S^{-1}\bar{\mathbf{v}}, \ldots, S^{-1}\bar{\mathbf{v}}_{10}$ and the additional point is $S^{-1}\bar{\mathbf{v}}_{11}$. The homogeneous coordinates of $S^{-1}\bar{\mathbf{v}}_{11}$ with respect to the frame $S^{-1}\bar{\mathbf{v}}, \ldots, S^{-1}\bar{\mathbf{v}}_{10}$ are the same as those of $\bar{\mathbf{v}}_{11}$ with respect to $\bar{\mathbf{v}}_1, \ldots, \bar{\mathbf{v}}_{10}$. We observe that with high probability, this invariant is perfect.

Observation 3. *Let $\phi \in ATF(\mathbb{F}_q^9)$ be a uniformly random form. Let $F(\phi, \bar{\mathbf{v}})$ be the invariant that outputs the homogeneous coordinates of $\bar{\mathbf{v}}_{11}$ with respect to the canonical frame $\bar{\mathbf{v}}_1, \ldots, \bar{\mathbf{v}}_{10}$ as described above. We observe that this invariant is well-defined with high probability if q is large enough. Moreover, with high probability, the invariant is perfect, in the sense that $F(\phi, \bar{\mathbf{v}}) = F(\phi, \bar{\mathbf{v}}')$ if and only if there is an automorphism of ϕ that sends $\bar{\mathbf{v}}$ to $\bar{\mathbf{v}}'$.*

7.4 Using F to solve the ATFE problem.

Now that we have a perfect invariant that is efficiently computable, we can instantiate the algorithm of Section 5.4. The idea is to compute lists L_1 containing pairs $(\bar{\mathbf{v}}, F(\phi_1, \bar{\mathbf{v}}))$, and L_2 containing $(\bar{\mathbf{u}}, F(\phi_2, \bar{\mathbf{u}}))$ where the $\bar{\mathbf{v}}$ and $\bar{\mathbf{u}}$ are rank-4 points for ϕ_1 and $F(\phi_2, \bar{\mathbf{u}}_i)$ respectively. We keep extending the lists until we have a collision $F(\phi_1, \bar{\mathbf{v}}) = F(\phi_2, \bar{\mathbf{u}})$, which happens after computing an expected number of $O(\sqrt{N_4})$ invariants. Since the invariant is perfect with high probability, we can assume that if a collision occurs, then there exists an isomorphism $S \in GL(\mathbb{F}_q^9)$ such that $\phi_2 = \phi_1 \circ S$ and $S\bar{\mathbf{u}} = \bar{\mathbf{v}}$, so given $\bar{\mathbf{v}}$ and $\bar{\mathbf{u}}$ we can efficiently find S with the system-solving approach of Section 4.

We can apply three optimizations: first, instead of computing the invariants for randomly chosen rank-4 points $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$, we first take $s = \lceil 2\log_2 q \rceil$ steps in the H_ϕ-graph. That is, we sample an rank-4 element $\bar{\mathbf{u}}$, but we compute the invariant for $H_\phi^s(\bar{\mathbf{u}})$. This ensures that we compute the invariant for points on the crater of one of the 2-volcanoes. If the number of rank-4 points is $N_4 = 2^d k$, then there are only k points on the craters, so we will find the first collision after computing in expectation only $O(\sqrt{k})$ invariants, a speedup of a factor $O(\sqrt{2^d})$.

Secondly, recall that to compute a single invariant we need to compute a chain $\bar{\mathbf{v}}, H_\phi^1(\bar{\mathbf{v}}), H_\phi^2(\bar{\mathbf{v}}), \ldots$ of length at least $s+11$. We can speed up the construction of the list L_1 by roughly a factor $s+11$, by first computing a chain of length $L >> s+11$ (e.g. $L = 20(s+11)$), and then extracting roughly $L - s - 11$ invariants from this long chain. We should only use this optimization for the list L_1. The probability that $F(\phi_2, H_\phi^s(\bar{\mathbf{v}}))$ collides with an invariant in L_1 is strongly

correlated with the probability that $F(\phi_2, H_\phi^{s+1}(\bar{\mathbf{v}}))$ results in a collision, so if we were to use the optimization also for L_2 we can no longer expect to find a collision after computing $O(\sqrt{k})$ invariants. Since extending the list L_1 is a factor $s+11$ cheaper than extending the list L_2, we get an optimal running time if we choose $|L_1| \approx (s+11)|L_2|$.

Finally, we can avoid the system-solving approach of [6]. If $S\bar{\mathbf{v}} = \bar{\mathbf{u}}$, then also $SH^i_{\phi_2}(\bar{\mathbf{v}}) = H^i_{\phi_1}(\bar{\mathbf{u}})$ for all i. Therefore we can just recompute the canonical projective frame for $\bar{\mathbf{v}}$ and $\bar{\mathbf{u}}$, and output the unique linear map that sends the frame of $\bar{\mathbf{u}}$ to the frame of $\bar{\mathbf{v}}$. This is simpler and more efficient than the system-solving approach of [6], but it does not make a significant difference to the overall cost of running the algorithm, since this last step was cheap compared to the cost of finding the collision $(\bar{\mathbf{u}}, \bar{\mathbf{v}})$.

The optimized algorithm goes as follows:

Input: Two isomorphic alternating trilinear forms $\phi_1, \phi_2 \in \mathrm{ATF}(\mathbb{F}_q^n)$.

Output: $S \in GL(\mathbb{F}_q^n)$ such that $\phi_2 = \phi_1 \circ S$.

0. Initialize empty lists L_1, L_2. Set $s = \lceil 2\log_2(q) \rceil$ and $L = 20(s+11)$.

1. Find a point $\bar{\mathbf{w}}_1$ of rank 6 for ϕ_1 e.g., by brute force or by solving a MinRank problem. Similarly, find $\bar{\mathbf{w}}_2$ of rank 6 for ϕ_2.

2. (Grow list L_1)

2a. Sample $\bar{\mathbf{u}}_0$ of rank 4 by walking in the G_{ϕ_1} graph: Set $\bar{\mathbf{w}}_1 \xleftarrow{\$} \mathrm{Rad}_{\phi_1}(\bar{\mathbf{w}}_1)$, and check if there exists $\bar{\mathbf{u}} \in \mathrm{Rad}_{\phi_1}(\bar{\mathbf{w}}_1)$ of rank 4 by solving a MinRank problem with 3 matrices and target rank 4. Repeat until a $\bar{\mathbf{u}}_0$ of rank 4 is found.

2b. Compute a chain $\bar{\mathbf{u}}_0, \bar{\mathbf{u}}_1 = H_{\phi_1}(\bar{\mathbf{u}}), \bar{\mathbf{u}}_2 = H^2_{\phi_1}(\bar{\mathbf{u}}), \ldots, \bar{\mathbf{u}}_L = H^L_{\phi_1}(\bar{\mathbf{u}})$ of length $L+1$.

2c. For i from s to $L-11$, compute $F(\phi_1, \bar{\mathbf{u}}_i)$, by extracting a projective frame from the sequence starting at $\bar{\mathbf{u}}_i$, and writing the next point in the sequence in homogeneous coordinates with respect to this frame. Add all the pairs $(\bar{\mathbf{u}}_i, F(\phi_1, \bar{\mathbf{u}}_i))$ to the list L_1.

3. (Grow list L_2)

3a. Sample $\bar{\mathbf{v}}$ of rank 4 by walking in the G_{ϕ_2} graph: Set $\bar{\mathbf{w}}_2 \xleftarrow{\$} \mathrm{Rad}_{\phi_1}(\bar{\mathbf{w}}_2)$, and check if there exists $\bar{\mathbf{v}} \in \mathrm{Rad}_{\phi_2}(\bar{\mathbf{w}}_2)$ of rank 4 by solving a MinRank problem with 3 matrices and target rank 4. Repeat until a $\bar{\mathbf{v}}$ of rank 4 is found.

3b. Compute $F(\phi_2, H^s_{\phi_2}(\bar{\mathbf{v}}))$ and add $(H^s_{\phi_2}(\bar{\mathbf{v}}), F(\phi_2, H^s_{\phi_2}(\bar{\mathbf{v}})))$ to the list L_2.

3c. Repeat step **3** until $|L_1| < (s+11)|L_2|$.

4. If there is a pair $(\bar{\mathbf{u}}, F(\phi_1, \bar{\mathbf{u}})) \in L_1$ and $(\bar{\mathbf{v}}, F(\phi_2, \bar{\mathbf{v}})) \in L_2$ such that $F(\phi_1, \bar{\mathbf{u}}) \sim F(\phi_2, \bar{\mathbf{v}})$ continue to step **5** otherwise go to step **2**.

5. Recompute the canonical projective frames $\bar{\mathbf{u}}_1, \ldots, \bar{\mathbf{u}}_{10}$ and $\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_{10}$ by iterating H_{ϕ_1} and H_{ϕ_2}. Finally, output the unique $S \in GL(\mathbb{F}_q^9)$ such that $S\bar{\mathbf{v}}_i = \bar{\mathbf{u}}_i$ for all i from 1 to 10.

Experiments. We implement the algorithm in `C++` and use it to solve random instances of the ATFE problem with the parameter set $n = 9, q = 524287$ proposed by [20], aiming for 128 bits of security level. For these instances, we can generate between 100 and 140 invariants per second for the list L_1, and between 3 and 4.5 invariants per second for L_2. If the number of rank-4 points is $N_4 = 2^d k$, then we are looking for collisions in a space of size approximately $k/|\mathrm{Aut}(\phi_1)|$ invariants[3]. So, we expect to find the first collision when

$$|L_1| \approx \sqrt{\frac{(s+11)k}{|\mathrm{Aut}(\phi)|}}, \text{ and } |L_2| \approx \sqrt{\frac{k}{(s+11)|\mathrm{Aut}(\phi)|}}.$$

The worst case solving time is when $|\mathrm{Aut}(\phi_1)|$ is small (2 is the smallest that we observed) and $N_4 \approx q^2$ odd. In that case, we would have an expected running time of 8 hours. We solved five ATFE problems and the solving time varied between 40 minutes and 4 hours. The large variability in solving time is to be expected. Some small amount of variability is due to the stochastic nature of the collision finding, sometimes we are lucky and find a collision early on, and sometimes we have to do more work, but most of the variability is due to the distribution of $k/|\mathrm{Aut}(\phi_1)|$. If ϕ_1 and ϕ_2 have more automorphisms (we observe that $|\mathrm{Aut}(\phi_1)|$ is usually between 2 and 6) or if the number of rank-4 points N_4 is divisible by a large power of 2, then the search space can be considerably smaller.

Breaking alternating trilinear form digital signatures. A public key of the [20] signature scheme consists of C isomorphic alternating trilinear forms $\phi_1, \cdots, \phi_C$, where for the $n = 9$ parameter set we have $C = 32$. The corresponding secret key consists of $C - 1$ isomorphisms connecting all the forms. To recover the secret key, an attacker needs to solve 31 ATFE problems. Naively solving each problem with our algorithm would take roughly between 31 times 40 minutes and 31 times 4 hours, i.e. between one and five days on a laptop. We can do better: To find the isomorphisms between a set of C forms $\phi_1, \ldots, \phi_C$, we first compute a list L_1 of rank-4 points and their invariants for ϕ_1 that is a factor $\sqrt{C-1}$ larger than optimal for a single execution of the ATFE-solving algorithm. Then, when we look for the isomorphism between ϕ_1 and ϕ_i for $i > 1$ the second list can be smaller by a factor $\sqrt{C-1}$. The total cost of finding the $C - 1$ isomorphisms is then only a factor $\sqrt{C-1}$ more expensive than solving an individual ATFE problem. With this strategy, we expect to be able to do a

[3] This is only approximate, because some of the points on the crater could be fixed points of some of the automorphisms.

full key recovery using between 4 and 22 hours, depending on the number of invariants on the craters.

To forge a single signature, we do not need to recover the entire secret key. We can start the signing procedure like an honest signer, then the attacker receives a set of challenges $b_1, \ldots, b_r \in [C]$ by hashing the message and the commitment $M|\psi_1|\ldots|\psi_r$. To finish the signature, the attacker only needs to solve the ATFE problem for the pairs of forms (ϕ_C, ϕ_{b_i}). In the $n = 9$ parameter set we have $r = 26$, so in the worst case the attacker needs to solve 26 ATFE problems, but this only happens if all the 26 challenges $b_1, \ldots, b_{26}$ are distinct and different from C. In general, the number of ATFE problems that an attacker needs to solve is $|\{b_1, \ldots, b_r, C\}| - 1$. For the $r = 26, c = 32$ parameter set the attacker can hash a few commitments (on average a few thousand) until it gets a set of challenges with $|\{b_1, \ldots, b_r, C\}| - 1 \leq 12$. So to forge a signature it suffices to solve only 12 simultaneous ATFE problems, which we expect to be able to do using between 2.5 and 14 hours.

Open problems

Our work shows that the ATFE parameters proposed by [20] are insecure, especially the $n = 9$ parameters. But there is still a lot of work to be done to understand the hardness of the ATFE problem. The most intriguing problem is to explain the observations about the rank-4 graphs in dimension 9. Why is there a group action from D_{N_4}, and what is the meaning of the H_ϕ function? Does this generalize to higher dimensions, and if so, can the structure be exploited to formulate more efficient algorithms for solving the ATFE problem in higher dimensions? Can the group action from D_{N_4} be used for constructive purposes? Instantiating the algorithm sketched in Section 5.4 is also left for future work.

Acknowledgements. We thank Simon-Phillip Merz, Luca De Feo, and Péter Kutas for the helpful discussions.

References

1. 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, Part I*, volume 12491 of *LNCS*, pages 507–536. Springer, Heidelberg, December 2020.
2. Ward Beullens. Not enough LESS: An improved algorithm for solving code equivalence problems over $\mathbb{F}_q$. In Orr Dunkelman, Michael J. Jacobson Jr., and Colin O'Flynn, editors, *SAC 2020*, volume 12804 of *LNCS*, pages 387–403. Springer, Heidelberg, October 21-23, 2020.
3. 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, Part II*, volume 12492 of *LNCS*, pages 464–492. Springer, Heidelberg, December 2020.

4. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 227–247. Springer, Heidelberg, December 2019.
5. Jean-François Biasse, Giacomo Micheli, Edoardo Persichetti, and Paolo Santini. LESS is more: Code-based signatures without syndromes. In Abderrahmane Nitaj and Amr M. Youssef, editors, *AFRICACRYPT 20*, volume 12174 of *LNCS*, pages 45–65. Springer, Heidelberg, July 2020.
6. Charles Bouillaguet, Jean-Charles Faugère, Pierre-Alain Fouque, and Ludovic Perret. Practical cryptanalysis of the identification scheme based on the isomorphism of polynomial with one secret problem. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 473–493. Springer, Heidelberg, March 2011.
7. 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, Heidelberg, May 2013.
8. Zhili Chen, Dung Hoang Duong, Ngoc Tuong Nguyen, Youming Qiao, Willy Susilo, and Gang Tang. On digital signatures based on isomorphism problems: QROM security and ring signatures. *Cryptology ePrint Archive*, 2022.
9. Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006. https://eprint.iacr.org/2006/291.
10. Giuseppe D'Alconzo and Andrea Gangemi. TRIFORS: Linkable trilinear forms ring signature. *Cryptology ePrint Archive*, 2022.
11. Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 759–789. Springer, Heidelberg, May 2019.
12. Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):691–729, 1991.
13. Joshua A Grochow and Youming Qiao. On the complexity of isomorphism problems for tensors, groups, and polynomials i: Tensor isomorphism-completeness. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
14. Jan Hora and Petr Pudlák. Classification of 8-dimensional trilinear alternating forms over GF(2). *Communications in Algebra*, 43(8):3459–3471, 2015.
15. Jan Hora and Petr Pudlák. Classification of 9-dimensional trilinear alternating forms over GF(2). *Finite Fields and Their Applications*, 70:101788, 2021.
16. Antonin Leroux and Maxime Roméas. Updatable encryption from group actions. *Cryptology ePrint Archive*, 2022.
17. Bernard Parisse and Renée De Graeve. *Giac/Xcas, version 1.9.0*, 2022. https://www-fourier.univ-grenoble-alpes.fr/~parisse/giac.html.
18. Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 33–48. Springer, Heidelberg, May 1996.
19. Andrew Sutherland. Isogeny volcanoes. *The Open Book Series*, 1(1):507–530, 2013.
20. 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 Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 582–612. Springer, Heidelberg, May / June 2022.
21. Gang Tang, Youming Qiao, and Joshua A Grochow. Average-case algorithms for testing isomorphism of polynomials, algebras, and multilinear forms. *journal of Groups, complexity, cryptology*, 14, 2022.
22. The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.7)*, 2022. https://www.sagemath.org.