Practical Post-Quantum Signature Schemes from Isomorphism Problems of Trilinear Forms

Gang Tang$^{1[0000-0002-1135-466X]}$, Dung Hoang Duong$^{2[0000-0001-8057-4060]}$, Antoine Joux$^{3[0000-0003-2682-6508]}$, Thomas Plantard$^{4[0000-0003-2521-2520]}$, Youming Qiao$^{1[0000-0003-4334-1449]}$, and Willy Susilo$^{2[0000-0002-1562-5105]}$

[1] Centre for Quantum Software and Information, School of Computer Science, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, Australia.
Youming.Qiao@uts.edu.au, gang.tang-1@student.uts.edu.au

[2] Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Northfields Avenue, Wollongong, NSW 2522, Australia.
{hduong,wsusilo}@uow.edu.au

[3] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany.
joux@cispa.de

[4] Emerging Technology Research Group, PayPal, San Jose, USA.
tplantard@paypal.com

Abstract. In this paper, we propose a practical signature scheme based on the alternating trilinear form equivalence problem. Our scheme is inspired by the Goldreich-Micali-Wigderson's zero-knowledge protocol for graph isomorphism, and can be served as an alternative candidate for the NIST's post-quantum digital signatures.

First, we present theoretical evidences to support its security, especially in the post-quantum cryptography context. The evidences are drawn from several research lines, including hidden subgroup problems, multivariate cryptography, cryptography based on group actions, the quantum random oracle model, and recent advances on isomorphism problems for algebraic structures in algorithms and complexity.

Second, we demonstrate its potential for practical uses. Based on algorithm studies, we propose concrete parameter choices, and then implement a prototype. One concrete scheme achieves 128 bit security with public key size $\approx$ 4100 bytes, signature size $\approx$ 6800 bytes, and running times (key generation, sign, verify) $\approx$ 0.8ms on a common laptop computer.

1 Introduction

Since the 1990s, several researchers observed the digital signature scheme obtained from the zero-knowledge proof protocol for graph isomorphism (GI) [41] by Goldreich, Micali and Wigderson (GMW), via the Fiat-Shamir transformation [39]. However, this scheme based on GI is not secure, as GI has long been considered as easy to solve in practice [62,63], not to mention Babai's quasipolynomial-time algorithm [6]. Still, this design pattern can be easily adapted to accommodate other isomorphism problems, and has been studied in multivariate cryptography and isogeny-based cryptography.

In multivariate cryptography, Patarin [71] first proposed to use polynomial isomorphism (PI) to replace graph isomorphism in the GMW identification protocol. Depending on the degrees and the number of polynomials involved, PI is actually a family of problems. The most studied cases include cubic forms and systems of quadratic polynomials. For systems of quadratic polynomials, there are also subcases such as homogeneous vs inhomogeneous (as explained in Example 3 of Section 4.2). Some problems, such as the isomorphism of quadratic polynomials with one secret, turn out to be easy [37,15,51]. The other proposal of Patarin in [71], namely utilizing hidden field equations, turns out to be more fruitful, as witnessed by the celebrated Rainbow scheme by Ding and Schmidt [28] which makes to the third round of the NIST call for proposals on post-quantum cryptography [3].

In isogeny-based cryptography, Couveignes [25] first proposed the use of class group actions on elliptic curves in cryptography. He adapted the GMW identification protocol to this action. Stolbunov [80] suggested to apply the Fiat-Shamir transformation to this identification protocol to get a signature scheme. However, the use of ordinary elliptic curves has issues including the subexponential-time quantum algorithm [23] and the slow performance. The attention then turned to *supersingular* elliptic curves [54], which lead Castryck, Lange, Martindale, Panne and Renes to propose the so-called CSIDH scheme based on a group action deduced from rational endomorphisms [22]. Since their work, there has been considerable progress on this signature scheme recently [38,11,31]. However, signature protocols based on class group actions met with several technical difficulties, such as computing the group action efficiently. Indeed, this was one key motivation to develop the SeaSign [38] and the CSI-FiSh [11] schemes. While these works come close to yield an efficient secure protocol based on class group actions, recent works [72,13] indicate that these parameters originally proposed in [22] do not achieve the claimed security level in the face of quantum attacks, leading to re-evaluations of those protocols; see [4] for more details.

These quantum attacks reaffirm the importance of quantum algorithms in post-quantum cryptography. Indeed, these attacks are based on careful analyses and clever uses of the quantum algorithms [72,13] for the dihedral Hidden Subgroup Problem (HSP) [57,74,58] and for the elliptic curve isogeny problem [23]. The Hidden Subgroup Problem (HSP) is one of the most prominent family of problems in quantum computation. HSP and the related hidden shift problem are of particular relevance to post-quantum cryptography. Generalizing Shor's quantum algorithms for integer factoring and discrete logarithm [79], they can also accommodate certain lattice problems [74] (the HSP for dihedral groups) and isogeny problems [23] (the abelian hidden shift problem).

In this paper, we consider the *alternating trilinear form equivalence* (ATFE) problem, defined as follows. Let $\mathbb{F}_q$ be the finite field of order q. A trilinear form $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ is *alternating*, if ϕ evaluates to 0 whenever two arguments are the same. Let A be an invertible matrix of size $n \times n$ over $\mathbb{F}_q$. Then A sends ϕ to another alternating trilinear form $\phi \circ A$, defined as $(\phi \circ A)(u, v, w) := \phi(A^{\mathrm{t}}(u), A^{\mathrm{t}}(v), A^{\mathrm{t}}(w))$. The ATFE problem then asks, given two alternating trilinear forms $\phi, \psi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$, whether there exists an invertible matrix A such that $\phi = \psi \circ A$.

ATFE can be formulated as an HSP instance over $\mathrm{GL}(n, q)$, the general linear group of degree n over $\mathbb{F}_q$. The research on HSP suggests that for $\mathrm{GL}(n, q)$ and symmetric groups, current quantum algorithm techniques cannot provide further speedup compared to classical algorithms [43,67,48]. This was termed by Moore, Russell, and Vazirani as "the strongest such insights we have about the limits of quantum algorithms" [68]. As far as we know, this insight had not been used to *directly* support the security of any practical post-quantum cryptosystems. In this paper, we will, for the first time, utilize this insight to investigate the practical use of ATFE in post-quantum cryptography.

Remark 1. Our use of HSP to support ATFE in post-quantum cryptography follows the use of HSP to support lattices in post-quantum cryptography. That is, by [74], certain lattice problems reduce to HSP over dihedral groups. However, to the best of our knowledge, it is not known that the HSP over dihedral groups reduces to lattice problems. Similarly, here ATFE can be formulated as a HSP over general linear groups, but the reverse direction is not known.

1.1 Theoretical preparations

Theoretical evidences for using ATFE in cryptography, besides HSP, are based on works from several research lines. Detailed discussions on the complexity, cryptography, and algorithm aspects of ATFE can be found in Sections 4 and 5. Here we present a brief summary illustrating some key ideas.

Recent advances in complexity theory [44,46] and algorithms [59,18,46] reveal a much clearer picture on the complexity of isomorphism problems of algebraic structures. In [46], it is shown that ATFE is complete for the *Tensor Isomorphism complexity class* (TI) [44]. This puts ATFE into a family of problems which have been studied in various areas including cryptography, machine learning, computer algebra, and quantum information. These problems are known to be difficult to solve *in practice*, a sharp contrast to graph isomorphism and code equivalence, which, despite being notoriously difficult for algorithms with rigorous worst-case analyses, allow for effective heuristic algorithms in practice [63,77].

The TI-completeness notion is useful in connecting ATFE with many algorithmic problems. However, as the reductions produce instances which are quite structured, from the algorithmic viewpoint it is most useful for worst-case analysis. For cryptographic uses of ATFE, average-case hardness or even stronger criteria are required. This is where *cryptography based on group actions* comes into the theme. Using group actions in cryptography has been studied by Brassard and Yung [17], Couveignes [25], and more recently in two papers [53,4], among others. In Section 4.2, we present evidence for ATFE to satisfy the one-way

[17] and pseudorandom [53] assumptions, which generalize the discrete logarithm hardness assumption and the Decisional Diffie-Hellman assumption, respectively.

To use ATFE in cryptography requires to pin down the algorithm for ATFE with the best time complexity. At first sight, ATFE seems little studied before. Fortunately, the Tensor Isomorphism-completeness of ATFE allows us to tap into years of research from multivariate cryptography, computational group theory, and theoretical computer science. This is because ATFE is polynomial-time equivalent to the cubic form isomorphism problem (CFI; see Definition 7), the quadratic form map isomorphism problem (QFMI; see Definition 8), and the class-2 p-group isomorphism problem (pGpI; see Definition 9). There is a large body of works in multivariate cryptography tackling CFI and QFMI, and in computational group theory and theoretical computer science tackling pGpI. The research in these communities produce non-trivial algorithms, utilizing tools and algorithmic ideas such as Gröbner basis, individualization and refinement, birthday paradox, and the min-rank attack. Still, these problems remain difficult to solve in practice.

1.2 Our contribution

In this paper, we propose and study a digital signature scheme based on the ATFE problem through the following steps.

1. We carefully study the hardness of ATFE and provide theoretical evidences for using ATFE in cryptography based on works from several research lines including hidden subgroup problems, multivariate cryptography, cryptography based on group actions, security proofs for cryptographic protocols in the quantum oracle model, and recent advances on isomorphism problems from algorithms and complexity.
2. We propose a post-quantum signature scheme based on the ATFE problem. Our scheme is inspired by the GMW zero-knowledge interaction protocol [41] for graph isomorphism. Our scheme is proven to be secure in the Random Oracle Model (ROM) based on the hardness of the ATFE problem.
3. We also provide some discussion and support for proving our schemes' security in the quantum random oracle model (QROM) based on [29,60]; see Section 3.2.
4. In Section 5, we go over many relevant algorithms from the study of CFI, QFMI, and pGpI and combine them with certain experiment studies on alternating trilinear forms, to pin down the best algorithm for ATFE to our best knowledge.
5. Based on the algorithmic study in Section 5, we propose criteria for setting the parameters of these schemes to achieve a fixed security level in Section 6.1. Further concrete instantiations lead to concrete schemes whose public key and signature sizes are reported in Table 1.
6. We implement a prototype of the basic scheme as in Table 1 using C, and report its preformance in Table 2. For more details see Section 6.2.

	Public key	Private key	Signature
Concrete Scheme 1	6384	6156	5018
Concrete Scheme 2	8160	6800	5542
Concrete Scheme 3	4080	3400	6816
Concrete Scheme 4	10560	7744	6309

Table 1. Output parameters for four concrete schemes based on ATFE for the 128-bit security. The sizes are measured in bytes.

7. Borrowing ideas from isogeny-based cryptography [38,11], we also observe a variant of this scheme with Merkle trees, which helps to reduce the public key size. See Appendix A for details.

	Set-Up	Sign	Verify
Concrete Scheme 1	285.9	471.7	416.5
Concrete Scheme 2	383.1	660.0	578.9
Concrete Scheme 3	190.7	795.4	708.8
Concrete Scheme 4	514.0	861.1	765.2

Table 2. Running times (in microsecond, μs, averaged over 10^5 runs) for Concrete Schemes 1 to 4 on Linux 5.11.0-37-generic with Intel Core i7-8565U CPU (1.80 GHz).

From the above, we belive that the main message is that this digital signature is potential for practical uses. In Section 1.3, we provide a comparison of our signature scheme with those are in the third round of NIST's Post-quantum Standardization process, and it shows that our scheme is comparable to those practical ones in terms of key sizes and running time. We also expect that our implementation can be be further improved and optimized, that we will leave as a future work. This scheme is also simple in both terms of conceptual and implementation viewpoints.

Our digital signature scheme can be served as an alternative candidate for post-quantum signatures. This also aligns to the recent announcement of NIST [69] at PQCrypto 2021 on calling for a general-purpose digital signature scheme which is not based on structured lattices, which are currently the most promising candidates of post-quantum signature standardization [3].

1.3 Comparison with 3rd round NIST's post-quantum signature schemes

Post-quantum cryptography has seen tremendous growth in the past few years. The National Institute of Standards and Technology (NIST) initiated the selection of proposals on post-quantum cryptographic algorithms for potential standardisation in November 2016, and the selection process came to the third round in mid 2020 [3]. There are three finalist proposals for signature schemes in the third round, namely Dilithium [7], Falcon [40], and Rainbow [78]. Dilithium and Falcon are based on lattice problems, and Rainbow is based on multivariate polynomials. Besides these schemes, the progress on isogeny-based signature schemes has been impressive in the past few years [38,11,27], and the SQISign scheme [27] probably comes closest to be practical.

We briefly review the public key and signature sizes of these schemes at NIST security level I or II, according to their latest specifications. The public key and signature sizes of Dilithium are 1312 bytes and 2420 bytes. The public key and signature sizes of Falcon-512 are 897 bytes and 666 bytes. Rainbow's signature size is small (528 *bits*), but it requires relatively large public key size ($\approx$ 58,800 bytes). The recently proposed SQISign achieves 204-byte signature size and 64-byte public key size. The running times of Dilithium, Falcon and Rainbow are very fast. As it would probably not be very instructive to list specific running times, we just mention that the running times of these schemes (sign and verify) are mostly within the range between 0.1ms and 1ms on a common laptop computer. On the other hand, the SQISign requires 2500ms for signing and 50ms for verification.

Based on the above observations, we believe that the parameters and running times of concrete schemes as shown in Tables 1 and 2 fall into the range for practical uses, when compared with the most promising schemes. Of course, Dilithium and Falcon take the lead in terms of these parameters. However, we offer as well a new direction with a *different* security basis and still a strong theoretical support.

1.4 On interactions with other research lines

From discussions above, this work has connections to many works from several research lines. We now provide some remarks to clarify the situations for readers with different backgrounds.

For experts on multivariate cryptography, we wish to deliver the message that Patarin's signature scheme based on polynomial isomorphism [71] could be practical if we are careful about the parameter choices, and replacing polynomial isomorphism with alternating trilinear form equivalence. Indeed, this scheme of Patarin was thought to be not practical, because the original parameters proposed were quickly broken [37,15,16]. Furthermore, some variants such as isomorphism of quadratic polynomials with one secret were shown to be easily solvable [15,14,10,51]. One main reason is that in the 2000's, signatures of say 3000 bytes would be considered as too large. Now in post-quantum cryptography, signatures of 5000 bytes are acceptable at least at the brainstorming stage. (For example, Dilithium produces signatures of 2420 bytes and SPHINCS+ [52], an alternate candidate in NIST round 3 [3], produces signatures of 7856 bytes.) With this in mind, it is actually reasonable to use cubic form isomorphism (CFI) to experiment with various parameters, assuming that the best algorithm runs in time say $q^n \cdot \mathrm{poly}(n, \log q)$ [15]. Utilizing ATFE has one advantage over CFI, as alternating trilinear forms require less storage than cubic forms ($\binom{n}{3}$ vs $\binom{n+2}{3}$), which results in better public key sizes.

For experts on isogeny-based cryptography, especially those who are familiar with SeaSign [38] and CSI-FiSh [11], s/he would quickly recognise that our scheme has the same structure. The key difference lies in using a different action. The class group action as in CSIDH [22] has smaller group and set element representations, but is more difficult to compute. The group action here (general linear groups acting on alternating trilinear forms) is easy to compute but the group and set elements are of larger sizes, resulting in larger public key and signature sizes.

Organization of the paper. In Section 2, we describe the alternating trilinear form equivalence problem (ATFE) and several variants. We describe the proposed schemes in Section 3, prove its security in the Random Oracle Model, and discuss its security in the Quantum Random Oracle Model (QROM). The Merkle variant is described in Appendix A. In Section 4, we discuss on the complexity and cryptography aspects of ATFE. In Section 5, we present a detailed study of algorithms for ATFE. In Section 6, we propose concrete parameters, describe our implementation, and report on its performance.

2 Preliminaries

2.1 Defining ATFE and variants

Our proposed signature protocol relies on the assumed hardness of the *alternating trilinear form equivalence* (ATFE) problem over finite fields. To define this problem we need some preparations.

Alternating trilinear forms with a natural group action. Let $\mathbb{F}_q$ be the finite field of order q. A trilinear form $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ is *alternating*, if ϕ evaluates to 0 whenever two arguments are the same. Let $\mathrm{ATF}(n, q)$ be the set of all alternating trilinear forms defined over $\mathbb{F}_q^n$. The general linear group $\mathrm{GL}(n, q)$ of degree n over $\mathbb{F}_q$ naturally acts on $\mathrm{ATF}(n, q)$ as follows: $A \in \mathrm{GL}(n, q)$ sends ϕ to $\phi \circ A$, defined as $(\phi \circ A)(u, v, w) := \phi(A^{\mathrm{t}}(u), A^{\mathrm{t}}(v), A^{\mathrm{t}}(w))$. This action defines an equivalence relation $\sim$ on $\mathrm{ATF}(n, q)$, namely $\phi \sim \psi$ if and only if there exists $A \in \mathrm{GL}(n, q)$, such that $\phi = \psi \circ A$.

Algorithmic representations. It is well-known that an alternating trilinear form $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ can be represented as $\sum_{1 \leq i < j < k \leq n} c_{i,j,k} e_i^* \wedge e_j^* \wedge e_k^*$, where $c_{i,j,k} \in \mathbb{F}_q$, e_i is the ith standard basis vector, e_i^* is the linear form sending $u = (u_1, \dots, u_n)^{\mathrm{t}} \in \mathbb{F}_q^n$ to u_i, and $\wedge$ denotes the wedge (or exterior) product. Indeed, we can view $e_i^* \wedge e_j^* \wedge e_k^*$ as an alternating trilinear form, sending (u, v, w), where $u = (u_1, \dots, u_n)^{\mathrm{t}}$, $v = (v_1, \dots, v_n)^{\mathrm{t}}$, $w = (w_1, \dots, w_n)^{\mathrm{t}}$ are in $\mathbb{F}_q^n$, to $\det \begin{bmatrix} u_i & v_i & w_i \\ u_j & v_j & w_j \\ u_k & v_k & w_k \end{bmatrix}$. Therefore, in algorithms we can store the alternating trilinear form ϕ as $(c_{i,j,k} : 1 \leq i < j < k \leq n)$, $c_{i,j,k} \in \mathbb{F}_q$, which requires $\binom{n}{3} \cdot \lceil \log q \rceil$ many bits.

The action of $\mathrm{GL}(n,q)$ on $\mathrm{ATF}(n,q)$ can be represented concretely as follows. Let $A = (a_{i,j}) \in \mathrm{GL}(n,q)$. It sends $e_i^* \wedge e_j^* \wedge e_k^*$ to $\sum_{1 \le r < s < t \le n} d_{r,s,t} e_r^* \wedge e_s^* \wedge e_t^*$, where $d_{r,s,t} = \det \begin{bmatrix} a_{i,r} & a_{i,s} & a_{i,t} \\ a_{j,r} & a_{j,s} & a_{j,t} \\ a_{k,r} & a_{k,s} & a_{k,t} \end{bmatrix}$. For general $\phi \in \mathrm{ATF}(n,q)$, the action of A can be obtained by linearly extending this action to each term $e_i^* \wedge e_j^* \wedge e_k^*$.

Formal statements of the algorithmic problems. We can now formally state the alternating trilinear form equivalence problem.

Definition 1. *The decision version of the alternating trilinear form equivalence problem (ATFE) is the following.*

Input *Two alternating trilinear forms $\phi, \psi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$.*
Output *"Yes" if there exists $A \in \mathrm{GL}(n,q)$ such that $\phi = \psi \circ A$. "No" otherwise.*

Definition 2. *The promised search version of the alternating trilinear form equivalence problem (psATFE) is the following.*

Input *Two alternating trilinear forms $\phi, \psi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$, with the promise that $\phi \sim \psi$.*
Output *Some $A \in \mathrm{GL}(n,q)$ such that $\phi = \psi \circ A$.*

Definition 3. *The promised search version of the alternating trilinear form equivalence problem with m-instances (m-psATFE) is the following.*

Input *m alternating trilinear forms $\phi_1, \ldots, \phi_m : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$, with the promise that $\phi_i \sim \phi_j$ for any $i, j \in [m]$.*
Output *Some $A \in \mathrm{GL}(n,q)$ and $i, j \in [m]$, $i \neq j$, such that $\phi_i = \phi_j \circ A$.*

Remark 2. It is not known whether the search version of ATFE reduces to the decision version in polynomial time. In [45], it was shown that for some related problems, such as the quadratic form map isomorphism (cf. Definition 8), search to decision can be done in time $q^{O(n)}$ (improving from $q^{n^2} \cdot \mathrm{poly}(n, \log q)$). So it is expected that for ATFE, a search to decision reduction can be achieved in time $q^{O(n)}$. However, a polynomial-time search to decision reduction seems difficult.

On the one hand, m-psATFE generalises the original version. On the other hand, it is easy to get a non-tight reduction from m-psATFE to the original version of psATFE. So we believe that m-psATFE is of the same difficulty as psATFE.

2.2 Digital signatures

Definition 4. *A signature scheme consists of a triplet of polynomial-time (possible probabilistic) algorithms* $(\textsc{KeyGen}, \textsc{Sign}, \textsc{Verify})$ *such that for every pair of outputs* $(\mathrm{PK}, \mathrm{SK}) \leftarrow \textsc{KeyGen}(1^\lambda)$ *and any n-bit message μ, we have*

$$\textsc{Verify}(\mathrm{PK}, \mu, \textsc{Sign}(\mathrm{SK}, \mu)) = 1$$

holds true, except with negligible probability (in λ).

A signature is said to be secure if it is impossible for an attacker to forge a valid signature. Explicitly, the standard definition of security for digital signature schemes are given in the game between the challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as the following.

- The challenger $\mathcal{C}$ generates $(\mathrm{PK}, \mathrm{SK}) \leftarrow \textsc{KeyGen}(1^{\lambda})$ and gives PK to $\mathcal{A}$.
- $\mathcal{A}$ is allowed to make the following queries at maximum Q times. For $i = 1, \cdots, Q$:
  - $\mathcal{A}$ chooses a message μ_i and sends to $\mathcal{C}$
  - $\mathcal{C}$ computes $\sigma_i \leftarrow \textsc{Sign}(\mathrm{SK}, \mu_i)$ and sends σ_i to $\mathcal{A}$.
- $\mathcal{A}$ outputs a forgery (μ^*, σ^*)
- $\mathcal{A}$ wins if $\textsc{Verify}(\mathrm{PK}, \mu^*, \sigma^*) = 1$ and $\mu^* \notin \{\mu_1, \cdots, \mu_Q\}$.

We say that a signature scheme is Existentially UnForgeable under adaptive Chosen Message Attacks (EUF-CMA) if no probabilistic polynomial-time adversary $\mathcal{A}$ wins the game above with non-negligible probability $\lambda^{-\mathcal{O}(1)}$.

3 Signature schemes based on ATFE

Our scheme is inspired by the zero-knowledge protocol for graph isomorphisms by Goldreich, Micali and Wigderson (GMW) [41]. As a high level, we will incorporate the ATFE to obtain a generalized GMW-like scheme and then apply the Fiat-Shamir transformation [39] to obtain a signature scheme. This basic scheme is described in Section 3.1. We emphasize that one may think it is straightforward to just replace the graph isomorphisms in GMW to ATFE, which is exactly the route we go, but the technical details are involved; see Section 3.1 for the detail. In Appendix A, we introduce a variant of the basic scheme in Section 3.1 by utilizing Merkle tree techniques that have been employed in many constructions including some isogeny-based signatures [38,11].

3.1 The basic scheme

The original GMW protocol [41] has two graphs as input. For the purpose of using it in identification and signature, it is useful to generalize this to more than two graphs, as already observed by several researchers including Patarin [71] and De Feo and Galbraith [38].

We present this slightly generalized scheme based on ATFE in Algorithms 1, 2, and 3. It involves four parameters: $n \in \mathbb{N}$ and a prime power q to specify $\mathrm{ATF}(n, q)$, the round number r, and the number of alternating trilinear forms in the public key $C = 2^c$. Note that we use $C = 2^c$ to simplify the analysis; in fact any number C would do.

Note that by randomly sampling $\phi \in \mathrm{ATF}(n, q)$, we sample independently randomly $\binom{n}{3}$ field elements from $\mathbb{F}_q$. By randomly sampling $A \in \mathrm{GL}(n, q)$, we can sample a random matrix from $M(n, q)$ until we get an invertible one, or use the method described in Section 6.2.

Algorithm 1: Key generation.

```
Input: The variable number $n \in \mathbb{N}$, a prime power $q$, the alternating trilinear
       form number $C = 2^c$.
Output: Public key: $C$ alternating trilinear forms $\phi_i \in \mathrm{ATF}(n, q)$ such that
        $\phi_i \sim \phi_j$ for any $i, j \in [C]$.
Private key: $C$ matrices $A_1, \ldots, A_C$, such that $\phi_i \circ A_i = \phi_C$.
Randomly sample an alternating trilinear form $\phi_C : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$.
Randomly sample $C - 1$ invertible matrices, $A_1, \ldots, A_{C-1} \in \mathrm{GL}(n, q)$.
For every $i \in [C - 1]$, $\phi_i \leftarrow \phi_C \circ A_i$.
For every $i \in [C - 1]$, $A_i \leftarrow A_i^{-1}$.
$A_C \leftarrow I_n$.
return Public key: $\phi_1, \phi_2, \ldots, \phi_C$. Private Key: $A_1, \ldots, A_C$.
```

Algorithm 2: Signing procedure.

```
Input: The public key $\phi_1, \ldots, \phi_C \in \mathrm{ATF}(n, q)$. The private key
       $A_1, \ldots, A_C \in \mathrm{GL}(n, q)$. $r \in \mathbb{N}$, $C = 2^c$. The message M. A hash
       function $H : \{0, 1\}^* \to \{0, 1\}^\ell$, with the promise that $\lfloor \ell/c \rfloor \geq r$.
Output: The signature $S$ on M.
for $i \in [r]$ do
    Randomly sample $B_i \in \mathrm{GL}(n, q)$.
    $\psi_i \leftarrow \phi_C \circ B_i$.
end
Compute $L = H(\mathrm{M}|\psi_1| \ldots |\psi_r) \in \{0, 1\}^\ell$.
/* For the next step we need $\lfloor \ell/c \rfloor \geq r$.                    */
Slice $L$ into $\lfloor \ell/c \rfloor$ bit strings in $\{0, 1\}^c$, and set $b_1, \ldots, b_r \in [C]$ to be the
  integer represented by the first $r$ bit strings.
for $i \in [r]$ do
    $D_i \leftarrow A_{b_i} B_i$. ;                    // Note that $\phi_{b_i} \circ D_i = \psi_i$.
end
return $S = (b_1, \ldots, b_r, D_1, \ldots, D_r)$.
```

It is straightforward to verify the correctness of the scheme. We now analyze its security. It is well-known that the Goldreich-Micali-Wigderson (GMW) protocol satisfies completeness, special soundness, and special honest-verifier zero knowledge properties. These allow us to prove the security of the digital signature scheme as follows.

Theorem 1. *The basic signature scheme described above is EUF-CMA secure in the Random Oracle Model (ROM) under the hardness of the m-psATFE problem.*

Proof. We proceed the proof by contradiction. Assume that there exists an adversary $\mathcal{A}$ that having maximum Q queries to the hash function H, which is modelled as random oracle, can break the EUF-CMA security, as described in Section 2.2, of the signature scheme. We will build an algorithm $\mathcal{B}$ that solves the ATFE with non-negligible probability using $\mathcal{A}$. The proof follows the standard one in Fiat-Shamir-type signature, we present it here for completeness.

Algorithm 3: Verification procedure.

```
Input: The public key $\phi_1, \ldots, \phi_C \in \mathrm{ATF}(n, q)$. The signature
       $S = (b_1, \ldots, b_r, D_1, \ldots, D_r)$, $b_i \in [C]$, $D_i \in \mathrm{GL}(n, q)$. The message M.
       The A hash function $H : \{0,1\}^* \to \{0,1\}^\ell$, with the promise that
       $\lfloor \ell/c \rfloor \geq r$.
Output: "Yes" if $S$ is a valid signature for M. "No" otherwise.
for $i \in [r]$ do
    Compute $\psi_i = \phi_{b_i} \circ D_i$.
end
Compute $L' = H(\mathrm{M}|\psi_1| \ldots |\psi_r) \in \{0,1\}^\ell$.
/* For the next step we need $\lfloor \ell/c \rfloor \geq r$.                    */
Slice $L'$ into $\lfloor \ell/c \rfloor$ bit strings in $\{0,1\}^c$, and set $b'_1, \ldots, b'_r \in [C]$ to be the
  integer represented by the first $r$ bit strings.
if for every $i \in [r]$, $b_i = b'_i$ then
    return Yes
else
    return No
```

At the beginning, $\mathcal{B}$ is given an instance of the C-psATFE problem, that are C alternative trilinear forms $\phi_1, \ldots, \phi_C : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ such that $\phi_i \sim \phi_j$ for any $i, j \in [C]$. The goal of $\mathcal{B}$ is to find $i \neq j$ and some $A \in \mathrm{GL}(n, q)$ such that $\phi_i = \phi_j \circ A$.

Let $L_1, \ldots, L_Q$ be random elements in $\{0,1\}^l$, which $\mathcal{B}$ will use to answer hash queries from the adversary $\mathcal{A}$, and let R be an entry from the set of possible random tapes of adversary $\mathcal{A}$. The algorithm $\mathcal{B}$ will take $(R, \phi_1, \ldots, \phi_C, L_1, \ldots, L_Q)$ as input. When $\mathcal{A}$ makes a signing query on the message M, then $\mathcal{B}$ executes the following steps:

- Take the next hash query value input to $\mathcal{B}$, and let this be L_j for $j \in [Q]$.
- Slice L_j into $\lfloor l/c \rfloor$ bit strings in $\{0,1\}^c$ and let $b_{j1}, \ldots, b_{jr} \in [C]$ be the integer represented by the first r bit strings of L_j.
- For $i \in [r]$, choose randomly $D_i \leftarrow \mathrm{GL}(n, q)$ and set $\psi_i := \phi_{b_{ji}} \circ D_i$.
- Define $L_j := H(M|\psi_1| \ldots |\psi_r)$. If this value has already been defined then we pick another values of D_i's.
- Return a signature $(b_{j1}, \ldots, b_{jr}, D_1, \ldots, D_r)$ to the adversary $\mathcal{A}$.

One can easily see that the distribution of the signature generated by $\mathcal{B}$ is statistically closed to that generated by the signing algorithm in Algorithm 2. In this case, the adversary $\mathcal{A}$ can verify the signature as in the verification procedure in Algorithm 3.

Assume now that $\mathcal{A}$ outputs a valid forgery $(b_1^*, \ldots, b_j^*, D_1^*, \ldots, D_r^*)$ for a message M^*. We let L^* be the corresponding hash query of the adversary, i.e., L^* is defined by $H(M^*|\psi_1^*| \cdots |\psi_r^*)$ by the algorithm $\mathcal{B}$. We let $(\psi_1^*, \cdots, \psi_r^*)$ be the associated commitments computed from $(b_1^*, \ldots, b_j^*, D_1^*, \ldots, D_r^*)$, i.e., $\psi_i^* = \phi_{b_i^*} \circ D_i^*$ for $i \in [r]$. Now the challenger $\mathcal{B}$ runs $\mathcal{A}$ a second time using the same randomness R as before. By the General Forking Lemma [12], $\mathcal{A}$ will output another forgery $(b'_1, \ldots, b'_j, D'_1, \ldots, D'_r)$ with associated commitments $(\psi'_1, \cdots, \psi'_r)$ for the same message M^* such that $\psi^*_i = \psi'_i$ for $i = 1, \cdots, r$ and $L^* \neq L'$, where L' is programmed to be $H(M^*|\psi'_1|\cdots|\psi'_r)$. Since $L^* \neq L'$, then there exist $i \in [r]$ such that $b^*_i \neq b'_i$. Now $\mathcal{B}$ outputs $A := D^*_i(D'_i)^{-1}$ as an answer for the given C-psATFE instance.

In fact, we have $\phi_{b^*_i} \circ A = \phi_{b^*_i} \circ D^*_i(D'_i)^{-1} = \psi^*_i \circ (D'_i)^{-1} = \psi'_i \circ (D'_i)^{-1} = \phi_{b'_i}$. Hence $\mathcal{B}$ already finds an invertible matrix $A \in \mathrm{GL}(n, q)$ and two indices $b^*_i \neq b'_i$ such that $\phi_{b^*_i} \circ A = \phi_{b'_i}$. This completes the proof. □

3.2 Security in Quantum Oracle Model (QROM)

Recently, the security of the Fiat-Shamir transformation in the quantum random oracle model (QROM) was established in [29,60]. Using these works, we follow the route of arguments in [11] to provide evidences – but not a rigorous proof – to support the security of our scheme in QROM. Indeed, to prove QROM security for concrete schemes based on Fiat-Shamir transformations is not trivial even based on [29,60]. For example, the QROM security of Dilithium, one finalist in the NIST call for proposals, is only proved assuming some conjecture [7].

In order to utilize [29,60], a key is to establish the collapsing property [60] or the quantum computational unique response property [29] of the GMW protocol for ATFE. This property is a generalization of the computational unique response property. In the context of the GMW protocol for ATFE, this property essentially asks, given $\phi, \psi \in \mathrm{ATF}(n, q)$ such that $\phi \sim \psi$, to produce different $A, B \in \mathrm{GL}(n, q)$ such that $\phi = \psi \circ A = \psi \circ B$, if there exist such A and B. Then note that AB^{-1} is a non-trivial automorphism of ψ. This then leads us to ask the following.

Definition 5 (Alternating trilinear form automorphism, ATFA). *Given a random $\phi \in \mathrm{ATF}(n, q)$, decide if the automorphism group $\mathrm{Aut}(\phi) = \{A \in \mathrm{GL}(n, q) : \phi \circ A = \phi\}$ is trivial, and if not, compute a non-trivial automorphism.*

The same problem for graph isomorphism, known as the graph automorphism problem, has received considerable attention [56]. From the worst-case analysis viewpoint, it was considered a difficult problem before Babai's breakthrough on graph isomorphism [6]. For random graphs, it is well-known that most graphs have the trivial automorphism group [32,82] as long as the number of edges is between $[cn, \binom{n}{2} - cn]$ for some constant c.

ATFA seems a difficult problem. The algorithm in [46] actually shows that for most $\phi \in \mathrm{ATF}(n, q)$, $|\mathrm{Aut}(\phi)| \leq q^{O(n)}$, but it runs in time $q^{O(n)}$. In [24,65,50], alternating trilinear forms in $\mathrm{ATF}(7, q)$ and $\mathrm{ATF}(8, q)$ over finite fields of characteristic not 3 are classified, and the automorphism groups are computed. To use such information, we will need to solve the ATFE problem for the alternating trilinear form at hand, and one of the canonical forms presented in [65]. As ATFE is considered as difficult (see Section 4.1), the classification information seems not very helpful, and this is only available for $n = 7$ or 8.

We believe it an interesting direction to explore ATFA further, and whether it is possible to prove formally the security of our protocol assuming that ATFA is hard, perhaps using the weakly collapsing property in [60].

4 Complexity and cryptography aspects of ATFE

4.1 ATFE in complexity theory

In Section 1.1, we mentioned the recent introduction of the Tensor Isomorphism-complete class (TI) in [44], which captures many isomorphism problems arising from multivariate crytography, machine learning, quantum information, and computer algebra. In [46], ATFE was proved to be TI-complete. Among those TI-complete problems, the following algorithmic problems are of particular relevance to our discussion.

Definition 6. *The 3-tensor isomorphism problem (3TI) is the following.*

Input *Two 3-way arrays $D = (d_{i,j,k}), E = (e_{i,j,k})$, where $d_{i,j,k}, e_{i,j,k} \in \mathbb{F}_q$ and $i, j, k \in [n]$.*

Output *"Yes" if there exist $A = (a_{i,r}), B = (b_{j,s}), C = (c_{k,t}) \in \mathrm{GL}(n, q)$, such that $D = (A, B, C) \star E$, where $(A, B, C) \star E := F = (f_{i,j,k})$, $f_{i,j,k} = \sum_{r,s,t\in[n]} a_{i,r}b_{j,s}c_{k,t}e_{r,s,t}$. "No" otherwise.*

3TI appears in quantum information, characterising equivalence classes of tripartite states under stochastic local operation and classical communication (SLOCC) [44].

Definition 7. *The cubic form isomorphism problem (CFI) is the following.*

Input *Two cubic forms (homogeneous degree-3 polynomials) $f, g \in \mathbb{F}_q[x_1, \dots, x_n]$.*

Output *"Yes" if there exists $A = (a_{i,j}) \in \mathrm{GL}(n, q)$, such that $f = A \star g$, where the action of A on g is by sending x_i to $\sum_{j\in[n]} a_{i,j}x_j$. "No" otherwise.*

CFI has been studied in multivariate cryptography [15] and theoretical computer science [1,2].

Definition 8. *The quadratic form map isomorphism problem (QFMI) is the following.*

Input *Two tuples of quadratic forms $\mathbf{f} = (f_1, \dots, f_m)$, $\mathbf{g} = (g_1, \dots, g_m)$, where $f_i, g_j \in \mathbb{F}_q[x_1, \dots, x_n]$ are quadratic forms (homogeneous degree-2 polynomials).*

Output *"Yes" if there exist $A = (a_{i,j}) \in \mathrm{GL}(n, q)$, $B = (b_{i,j}) \in \mathrm{GL}(m, q)$, such that $\forall i \in [m]$, $f'_i = A \star g_i$, where $f'_i = \sum_{j\in[m]} b_{i,j}f_j$, and the action of A on g_i is by sending x_i to $\sum_{j\in[n]} a_{i,j}x_j$. "No" otherwise.*

QFMI has been studied in multivariate cryptography. It was first raised by Patarin [71] and has been studied in several works including [37,16,10]. Several variants of this problem have also been studied, such as replacing quadratic forms with quadratic polynomials (from homogeneous to possibly inhomogeneous), or restricting B to be the identity matrix (also known as the one secret version of the problem).

Definition 9. *The class-2 and exponent-p p-group isomorphism problem (pGpI) is the following.*

Input *Two sets of matrices* $A = \{A_1, \ldots, A_m\}, B = \{B_1, \ldots, B_m\} \in \mathrm{GL}(n, p)$, *with the promise that A (resp. B) generates a p-group G (resp. H) of class 2 and exponent p.*

Output *"Yes" if G and H are isomorphic (as abstract groups). "No" otherwise.*

pGpI has long been known to be one bottleneck case of the group isomorphism problem, which asks whether two finite groups are isomorphic. It is studied in both computational group theory [70,81,19] and theoretical computer science [59,18,44].

The following theorem is important for our understanding of ATFE.

Theorem 2 ([44,46]). *The following problems are equivalent under polynomial-time reductions: ATFE, 3TI, CFI, QFMI, and pGpI.*

Theorem 2 allows us to tap into research areas such as multivariate cryptography, computational group theory, and theoretical computer science, to understand the complexity of ATFE. In particular, we have seen that CFI and QFMI are known to be difficult in multivariate cryptography, and pGpI is known to be difficult in computational group theory. This gives us confidence in the worst-case hardness of ATFE. However, for cryptographic uses, ATFE needs to be difficult in the average-case sense. This is addressed in the next subsection.

4.2 ATFE and cryptography based on group actions

Let G be a group and S a set. A group action is a function $\alpha : G \times S \to S$ satisfying certain axioms. For the purpose of this article we don't need to spell out these axioms; instead, it is enough to realize that the functions underlying isomorphism problems are all group actions.

Cryptography based on group actions, as a framework, has been studied by Brassard and Yung [17], Couveignes [25], and more recently in two papers [53,4]. We review this framework and explain the roles of the discrete logarithm problem and ATFE in this framework.

In [17], Brassard and Yung defined the group action α to be *one-way*, if there exists $s \in S$, such that $\alpha_s : G \to S$, defined as $\alpha_s(g) = \alpha(g, s)$, is a one-way function. In [53], this is slightly relaxed to α_s is a one-way function for a random $s \in S$. The following example, known at least since [25], shows how to interpret the discrete logarithm problem as a problem about group action.

Example 1. To illustrate the notion of one-way group actions, let us consider an important group action in cryptography. Let C_p be the cyclic group of order p, and let $\mathrm{Aut}(C_p)$ be the automorphism group of C_p. Note that $G = \mathrm{Aut}(C_p) \cong \mathbb{Z}_p^*$, the multiplicative group of units in $\mathbb{Z}_p$. Then given $a \in \mathbb{Z}_p^*$ and $g \in C_p$, a sends g to g^a. Let $S = C_p \setminus \{\mathrm{id}\}$ where id is the identity element, and let $\alpha : \mathrm{Aut}(C_p) \times S \to S$ be the group action just defined. Then α is one-way, if and only if α_g is one-way for some $g \in S$, if and only if the discrete logarithm problem (with a fixed generator) is one-way.

Clearly, the action underlying ATFE being one-way in the relaxed sense is equivalent to saying that the problem of solving psATFE is hard on average.

In [25], Couveignes studied what he called hard homogeneous spaces, which is in fact also a group action with certain properties. In particular, he defined the parallelization problem for a group action α as follows. Given $s_1, t_1, s_2 \in S$ with the promise that there exists $g \in G$ such that $\alpha(g, s_1) = t_1$, compute $\alpha(g, s_2)$. For the group action defining discrete logarithm as in Example 1, its parallelization problem is hard on average is equivalent to the Computational Diffie-Hellman assumption.

Recently, the notion pseudorandom group actions was independently introduced in [53] and [4].[5] Briefly speaking, a group action $\alpha : G \times S \to S$ is pseudorandom, if efficient algorithms cannot distinguish the following two distributions. The first distribution is the random distribution, namely $(s, t) \in S \times S$ where $s, t \in_R S$. The second distribution is the pseudorandom distribution, namely $(s, t) \in S \times S$ where $s \in_R S$, and $t = \alpha(g, s)$ where $g \in_R G$. In [53], it was observed that this assumption generalizes the Decisional Diffie-Hellman assumption. We reproduce this example here.

Example 2. Let C_p, $G = \mathrm{Aut}(C_p)$, and $S = C_p \setminus \{\mathrm{id}\}$ be from Example 1. Note that the action of G on C_p is transitive, i.e. for any $g, h \in S$, there exists $a \in G$ such that $g^a = h$. In particular, for a fixed $g \in S$, when a is uniformly sampled from G, g^a is uniformly sampled from S. Let G act on $S \times S$ diagonally, i.e. $a \in G$ sends (g, h) to (g^a, h^a). Then the random distribution (of this diagonal action) is $((g, h), (g', h')) = ((g, g^a), (g^b, g^c))$ where $g \in_R S$, $a, b, c \in_R G$. The pseudorandom distribution is $((g, h), (g^b, h^b)) = ((g, g^a), (g^b, g^{ab}))$ where $g \in_R S$ and $a, b \in_R G$. Distinguishing these two distributions is then exactly the Decisional Diffie-Hellman problem.

We give an example suggesting that the pseudorandom group action is a useful criterion for cryptographic uses in the context of multivariate cryptography as follows.

Example 3. Consider the quadratic form map isomorphism problem (QFMI) from Definition 8, where $\mathrm{GL}(n, q) \times \mathrm{GL}(m, q)$ acts on tuples of quadratic forms $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{F}_q[x_1, \ldots, x_n]$. Consider the following two variations. First, we relax f_i to be quadratic polynomials, that is, f_i's are allowed to have linear and constant terms. Call this Variant 1 of QFMI. Second, we relax f_i's to be

[5] In [4] this is called weak pseudorandom group actions.

quadratic polynomials with constant terms being 0, that is, f_i's are allowed to have linear terms but no constant terms. Call this Variant 2 of QFMI.

The experience in multivariate cryptography (cf. Bouillaguet's thesis [14]) suggests that Variant 2 is easier than Variant 1, which is in turn easier than QFMI itself. From the pseudorandom group action viewpoint, Variant 1 is clearly not pseudorandom, as the constant terms are not changed under the group action. Variant 2 is also not pseudorandom: in the setting $m = n$ (the most studied situation), the rank of the n linear forms from f_i's is an invariant under the group action, which can be computed easily to distinguish the random and pseudorandom distributions. (Note that over $\mathbb{F}_q$, the rank of n linear forms in n variables is not full with probability $\geq 1/q^{\Theta(1)}$.)

It is clear that the pseudorandom assumption is stronger than the one-way assumption and the assumption that solving paralleization is hard. In [53,4], pseudorandom group actions are shown to have applications ranging from pseudorandom functions, to signature, and to oblivious transfer. The candidate pseudorandom group actions are the 3-tensor action as in Definition 6 (proposed in [53]) and the class group action underlying CSIDH [22] (proposed in [4]). Note that certain technical modifications are required to address some computational issues in the class group action underlying CSIDH. Furthermore, certain applications of pseudorandom group actions in [4] require the group to be commutative.

The main conjecture in this article is the following.

Conjecture 1. The group action underlying ATFE is pseudorandom.

To prove ATFE to be pseudorandom (even based on certain assumptions) seems difficult. Instead, as customary for this type of question, we provide certain arguments to support Conjecture 1.

- Several researchers have noted that the mathematics of alternating trilinear forms is "much harder" [5], or "much more complicated (and interesting)" [30], especially when compared to alternating bilinear forms. For example, in general one cannot expect to classify alternating trilinear forms when n is large enough.
- A basic approach to refute an action from being pseudorandom is to identify easy-to-compute isomorphism invariants, which are quantities unchanged by the group action. Such isomorphism invariants are also expected to be non-trivial for random instances. For example, rank is an isomorphism invariant for the action of $\mathrm{GL}(n, q) \times \mathrm{GL}(n, q)$ on $M(n, q)$ by left and right multiplications. It is non-trivial because at least $1/q^{\Theta(1)}$ fraction of $M(n, q)$ are of non-full rank.

  As far as we known, for 3TI, CFI, QFMI, and pGpI, ATFE, despite having been studied in several areas for decades, no such isomorphism invariants are found. For example, tensor rank is certainly an isomorphism invariant for 3TI, but it is NP-hard [49], and most tensors are of full-rank, which makes it not useful for breaking the pseudorandom assumption.
- There are some non-trivial attack strategies in [53] supporting 3TI to be pseudorandom, including utilizing supergroups and invariant theory. These

attack strategies works for certain settings (such as unitary groups and special linear groups), but do not work with general linear groups. Such arguments can be used to support Conjecture 1 as well.

5 Algorithms for ATFE

In this section we study algorithms for ATFE, which are crucial to pin down the parameter choices of our signature scheme. Based on these, we state the best algorithm for ATFE, based on the current literature, runs in time

$$O\big(q^{2/3\cdot n}\cdot n^{2\cdot\omega}\cdot\log_2(q)\big), \tag{1}$$

where ω is the matrix multiplication exponent. This is based on the heuristic (but still analysable) algorithm to be explained below.

Let us first list known algorithms with rigorous analyses for ATFE. It should be noted that current algorithms actually solve the search version of ATFE, so the following applies to psATFE as well.

- The brute-force algorithm for ATFE enumerates $A \in \mathrm{GL}(n,q)$ and then verifies if $\phi = \psi \circ A$. This runs in time $q^{n^2}\cdot\mathrm{poly}(n,\log q)$.
- Worst-case algorithms: One can adapt the dynamic programming method in [59] to obtain an algorithm for ATFE in time $q^{\frac{1}{4}n^2+O(n)}$.
- Average-case algorithms: An average-case algorithm for ATFE needs to solve the problem for inputs of the form ϕ, ψ for most $\phi \in \mathrm{ATF}(n,q)$ and arbitrary $\psi \in \mathrm{ATF}(n,q)$. In [46], an average-case algorithm for ATFE in time $q^{O(n)}$ was presented, which works for all but $\frac{1}{q^{\Omega(n)}}$ fraction of $\phi \in \mathrm{ATF}(n,q)$.[6] However, the constant hidden in the big O is at least 4 [46, arXiv version, Appendix C], therefore it is still not useful in practice.
- Quantum algorithms: as described in Section 1, known research on the hidden subgroup problem over general linear groups has mostly produced negative evidences. This may partly explain why there seems no non-trivial quantum algorithms in this setting, unlike the dihedral hidden subgroup problem. Still, it may be possible to use Grover search [47] to help speeding up some procedures useful for solving ATFE; see Remark 5.

From the above, we see that average-case algorithms are known to run in time $q^{O(n)}$. For practical purposes, however, it is desirable to obtain an algorithm in time $q^n\cdot\mathrm{poly}(n,\log q)$. The main purpose of this section is to explore ways of developing *heuristic* algorithms in time $q^{2/3\cdot n}\cdot\mathrm{poly}(n,\log q)$. This seems to be a previously uncharted territory of heuristic algorithms for ATFE, but, thanks to Theorem 2, we can borrow ideas and experiences from multivariate cryptography and computational group theory. Conversely, any progress on ATFE should help with making progress on CFI and pGpI that have withstood attacks for decades.

[6] In [46] an algorithm in such time was presented for CFI, but its algorithmic idea can be readily applied to ATFE.

5.1 A useful isomorphism invariant

To start with, we introduce the following notion which will be important for discussions on heuristic algorithms.

Definition 10. *Let $\phi \in \mathrm{ATF}(n,q)$. For $u \in \mathbb{F}_q^n$, let $\phi_u : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ be the alternating bilinear form defined by $\phi_u(v,w) = \phi(u,v,w)$.*

The rank of ϕ_u as an alternating bilinear form is an important invariant under isomorphism. That is, if $\phi = \psi \circ A$, then A must sends $u \in \mathbb{F}_q^n$ with $\mathrm{rk}(\phi_u) = r$ to some $v \in \mathbb{F}_q^n$ with $\mathrm{rk}(\psi_v) = r$. Also note that the rank of an alternating bilinear form must be an even number.

Given $\phi \in \mathrm{ATF}(n,q)$ and $r \in \mathbb{N}$, let $R_{\phi,r} := \{u \in \mathbb{F}_q^n : \mathrm{rk}(\phi_u) = r\}$. The sizes of $R_{\phi,r}$, $r \in \mathbb{N}$, form an important isomorphism invariant of ϕ. At present, there seems little research into the sizes of R_ϕ. So we run experiments to get an idea for small n and q. Our experiment results are summarized in Table 3.

	2	4	6	8	10
$(7,5)_{100}$	5.76	16218.24	61900	N/A	N/A
$(8,5)_1$	0	2064	388020	N/A	N/A
$(8,7)_1$	0	17100	5747700	N/A	N/A
$(9,2)_{100}$	0.01	9.77	281.62	291.60	N/A
$(9,3)_{100}$	0	30	7064.24	12587.76	N/A
$(9,5)_1$	0	216	409908	154300	N/A
$(10,3)_{100}$	0	0.96	2451.74	56595.3	N/A
$(12,3)_1$	0	0	10	25312	50918

Table 3. Experiment results on rank statistics. The first columns are of the form $(n,q)_a$, where a denotes the number of experiments run. The first row is the value of r. The results are averaged over these many experiments. All these experiments produce ϕ with $R_{\phi,0} = \{\mathbf{0}\}$.

Remark 3. Some interesting observations can be made regarding Table 3. For odd $n = 2k+1$ and random $\phi \in \mathrm{ATF}(n,q)$, it seems that $|R_{\phi,2(k-1)}|$ is slightly larger than q^{n-1}, and $|R_{\phi,2(k-2)}|$ is slightly larger than q^{n-6}. For even $n = 2k$ and random $\phi \in \mathrm{ATF}(n,q)$, it seems that $|R_{\phi,2(k-2)}|$ is slightly larger than q^{n-3}, and $|R_{\phi,2(k-3)}|$ is slightly larger than q^{n-10}. It would be interesting to formally prove certain properties of rank statistics for random $\phi \in \mathrm{ATF}(n,q)$.

Remark 4. Rank statistics has been used to tackle $p\mathsf{GpI}$ in [18]. Following the approach in [18], one can design a heuristic algorithm in time $q^n \cdot \mathrm{poly}(n, \log q)$. However, that approach requires to enumerate all low rank matrices, which seems to prohibit from getting an algorithm in time $q^{cn} \cdot \mathrm{poly}(n, \log q)$, $c < 1$.

5.2 The heuristic algorithm with running time in Equation (1)

We now describe the heuristic algorithm in time $O\big(q^{2/3\cdot n} \cdot n^{2\cdot\omega} \cdot \log_2(q)\big)$. This algorithm closely follows the algorithm for QFMI in [16] based on the Gröbner basis method.

Setting up the equations. One approach to set up a system of polynomials to solve ATFE is as follows. Let $\phi, \psi \in \mathrm{ATF}(n, q)$. We set up two $n \times n$ variable matrices X and Y, and impose that $XY = I$, where I is the identity matrix. (So X and Y are inverses to each other.) This gives us n^2 quadratic equations. Then instead of setting up equations in the form $\phi(X(u), X(v), X(w)) = \psi(u, v, w)$, we set it as

$$\phi(X(u), X(v), w) = \psi(u, v, Y(w)). \tag{2}$$

This helps us to get quadratic equations instead of cubic ones. This gives us $\binom{n}{3}$ equations. In total, we have $\binom{n}{3} + n^2$ quadratic equations in $2n^2$ variables.

The direct Gröbner basis method: experimental results. To directly solve the above system of polynomial equations seems difficult. This echoes the experiences in directly solving QFMI and CFI by the Gröbner basis method in [15,16,14]. For example, in [14, Chap. 16] it was reported that for CFI, Gröbner basis attacks succeeded for $n = 6$ (in about 400 seconds) and $n = 7$ (in about 17000 seconds), but failed for $n = 8$.

We experimented with this direct attack for ATFE on a workstation[7] using Maple [61]. We carried out successful computations for $n = 6$ and $q = 5$ in about 700 seconds, but could not achieve a successful one for $n = 7$ and $q = 5$ (using several monomial orders and both the quadratic and cubic methods): our experiments suggested that the memory usage went beyond 87 GB.

Maybe surprisingly, the Gröbner basis attack can be improved by adding more, seemingly redundant equations to the system. More precisely, we add equations to encode $YX = I$ on top of $XY = I$, and complete $\phi(X(u), X(v), w) = \psi(u, v, Y(w))$ with equations encoding $\phi(X(u), v, w) = \psi(u, Y(v), Y(w))$, $\phi(u, v, w) = \psi(Y(u), Y(v), Y(w))$, and $\phi(X(u), X(v), X(w)) = \psi(u, v, w)$. On top of this, we fix a subset of the variables and randomly assign values to them, before calling the Gröbner basis algorithm. This is in the spirit of hybrid Gröbner basis algorithms and also exploit the fact that many solutions may exist, especially in small dimension. This leads to a very fast attack on $n = 7$ and permits breaking $n = 8$. However, $n = 9$ remains out of range of this improvement.

Since the behaviours of Gröbner basis algorithms in our case on small sizes seemed not uniform, we didn't extrapolate the time needed by Gröbner basis attacks. Instead, we performed a theoretical anlysis as follows.

The direct Gröbner basis method: theoretical asymptotic analyses. From the theoretical analysis side, while a rigorous analysis of the Gröbner basis algorithm is notoriously difficult, it is possible to give some estimates based on certain

[7] Processor: 2.6 GHz 18-core Intel(R) Xeon(R) Gold 6132; Memory 87 GB.

assumptions. [8] Following the approach in [16,14], we can give an upper bound on the regularity for these equations, *assuming* that these polynomials form a semi-regular sequence and using the results of Bardet et al. [8,9].

Let $(f_1, \dots, f_m)$, $f_i \in \mathbb{F}[x_1, \dots, x_n]$, be a sequence of polynomials. Generalising the classical notion of regular sequences of polynomials (for $m = n$), Bardet and Faugère introduced the notion of semi-regular sequences of polynomials (for $m > n$), and Bardet et al. studied the degree of regularities of the ideal spanned by these polynomials in [8,9]. An asymptotic estimate on the degree of regularity for semi-regular sequences of $m = \alpha N$ *quadratic* polynomials in N variables, α a constant, is given in [9] as

$$N\big(\alpha-\frac{1}{2}-\sqrt{\alpha(\alpha-1)}\big)-\frac{a_1}{2(\alpha(\alpha-1))^{1/6}}\cdot N^{1/3}-\big(2-\frac{2\alpha-1}{4\sqrt{\alpha(\alpha-1)}}\big)+O(1/N^{1/3}), \tag{3}$$

where $a_1 \approx -2.33811$.

To apply Eq. (3) to ATFE, we need to *assume* that (i) the equations form a semi-regular sequence, and (ii) Eq. (3) applies to the setting when α is not necessarily a constant. These were assumed for QFMI and CFI in [16,14], and we assume so here as well. By Eq. (2), we have $\binom{n}{3} + n^2$ quadratic equations in $2n^2$ variables. Using Eq. (3) (with $\alpha = \frac{n}{12} + \frac{1}{4} + \frac{1}{6n}$ and $N = 2n^2$), we obtain that the degree of regularity is asymptotically $3n$. Therefore the F5 algorithm of Faugère [33] is expected to perform

$$O(N^{\omega\cdot 3n}) = O(2^{6\omega\cdot n\cdot\log_2(n)}) \tag{4}$$

many arithmetic operations.

We would like to emphasize that the above estimation is heuristic, and only gives an upper bound. So while it is useful as a guidance, it is necessary to perform experiments.

The XL method: theoretical asymptotic analyses. Another approach of solving system of polynomial equations is the eXtended Linearisation (XL) method [21]. The XL method with a sparse matrix (such as Wiedemann) solver was proposed in [83] as an alternative to F4/F5 [33]. For the purpose of theoretical analyses here, the sparse matrix solver could improve the matrix multiplication exponent ω to 2 in Equation (4). In particular, this implies that we need to take into account n^{12} in determining the underlying finite field order q. Because of this, for the sake of security, we will use 2 instead of ω in the choice of parameters (cf. Section 6.1). It will be interesting to experiment with XL for solving the polynomial equation system in Equation (2), and we leave this in a future work.

Gröbner basis attack with partial information. To make progress, we can set the first row of the variable matrix X to be known, which amounts to say that the image of e_1 under X is known. This also gives a linear constraint on the

[8] We would like to thank Charles Bouillaguet for his help with understanding these methods here.

columns of Y. We then essentially solve a system of polynomials in $2(n^2 - n)$ many variables. Such ideas have been used to solve CFI in [15].

We experiment with the improved Gröbner basis method with guessing one row (but without fixing a subset of variables and randomly assigning values to them). It turns out that, in this setting, the maximum degrees for the Gröbner basis computation to succeed are 3 for n up to 13. Furthermore, we can even use results from the Gröbner basis computation at degree 2 to solve the system. We take these as evidence that the Gröbner basis with partial information runs much faster than the method without partial information, and make the following assumption.

Assumption 1. Suppose $\phi, \psi \in \mathrm{ATF}(n, q)$ such that $\phi = \psi \circ A$. Suppose the first column of A is known. Then the Gröbner basis computation reveals the rest entries of A in time $O(n^{2 \cdot \omega} \cdot \log(q))$.

Assumption 1 is consistent with the discoveries in [37] that the inhomogeneous version of QFMI can be solved by the Gröbner basis method in time $O(n^9)$, and in [15] that CFI can be solved with the Gröbner basis method with partial information in time $O(n^6)$.

Making use of the birthday paradox. We now combine Assumption 1 with birthday paradox, following the idea in [16][9]. The key idea is to make use of those u whose ϕ_u has low rank.

Let us describe the idea in a concrete setting. Let $\phi, \psi \in \mathrm{ATF}(n, q)$. For $u \in \mathbb{F}_q^n$, ϕ_u is the bilinear form defined in Definition 10, and *assume* that there exists $r \in \mathbb{N}$, such that $|R_{\phi,r}| \approx q^{2n/3}$. Then we sample $q^{n/3}$ vectors, say $S \subseteq R_{\phi,r}$ and $T \subseteq R_{\psi,r}$ respectively, in time $O(q^{2n/3})$. By the birthday paradox, there exist $u \in S$ and $v \in T$, such that $Au = v$ with constant probability. Knowing the image of A on one vector should help with recovering the whole A by using e.g. the Gröbner basis attack with partial information.

Note that the above relies on the assumption[10] that there exists r such that $|R_{\phi,r}| \approx q^{2n/3}$. For other parameters this method still gives improvement but, because it needs to balance sampling vectors to get S and T and the use of birtday paradox, so $q^{2/3 \cdot n}$ is the best it can achieve.

Combining Assumption 1 with the above, we have a heuristic algorithm in time $O\big(q^{2/3 \cdot n} \cdot n^{2 \cdot \omega} \cdot \log_2(q)\big)$.

5.3 Attacks based on min-rank

Several attacks on the hidden field equation proposal [71] rely on solving the min-rank problem: given matrices $A_0, A_1, \ldots, A_m \in M(n, q)$, let $\mathcal{A} = \{A_0 +$

[9] There is another algorithm in [16] with time complexity $q^{n/2} \cdot \mathrm{poly}(n, \log q)$, but it was designed for characteristic 2 fields, which we do not use for concrete instantiations in Section 6.1.

[10] By Table 3, for $n = 10$ which will be used to instantiate our scheme in Section 6.1, we see that $|R_{\phi,6}| \approx q^7$, where $7 \approx 20/3$. On the other hand for odd n there is no such r.

$\sum_{i=1}^{n} \alpha_i A_i : \alpha_i \in \mathbb{F}_q\}$ be the affine subspace of $M(n, q)$. The min-rank problem asks to compute a matrix of minimum rank in $\mathcal{A}$. It is NP-complete [20], but several non-trivial algorithms have been developed for this problem.

The min-rank problem is of relevance to ATFE because of the rank statistics in Section 5.1. To make the connection more obvious, we review the well-known connection between alternating trilinear forms and 3-way arrays.

Definition 11. *Given $\phi \in \mathrm{ATF}(n, q)$, written as $\phi = \sum_{1 \le i < j < k \le n} a_{i,j,k} e_i^* \wedge e_j^* \wedge e_k^*$ (cf. Section 2.1), we define a 3-way array $(b_{r,s,t})_{r,s,t \in [n]}$, $b_{r,s,t} \in \mathbb{F}_q$, by setting $b_{r,s,t} = \phi(e_r, e_s, e_t)$. Then for $t \in [n]$, construct a matrix $B_t = (b_{r,s,t})_{r,s \in [n]}$. Let $u = (u_1, \dots, u_n)^t \in \mathbb{F}_q^n$. Then it is easy to verify that $\mathrm{rk}(\phi_u) = \mathrm{rk}(\sum_{i=1}^{n} u_i B_i)$.*

As we have seen in Section 5.2, the rank statistics is useful for algorithmic purposes. In particular, in the algorithm in Section 5.2, it is useful to get some $u \in \mathbb{F}_q^n$ of such that ϕ_u is of minimum rank among non-zero vectors. Note though that for isomorphism testing purposes, just one such u seems not quite helpful. Instead, we need a large fraction, if not all, of the u whose ϕ_u is of minimum rank in order to create a collision.

So we examine some algorithms for the min-rank problem and their potential consequences on ATFE. Note that the instance of the min-rank problem relevant to us is the following: given $A_1, \dots, A_n \in M(n, q)$ be n matrices, and $\mathcal{A}$ be the linear subspace of $M(n, q)$ spanned by A_i's. The problem is to find a non-zero matrix $A \in \mathcal{A}$ of rank $\le r$.

The kernel attack. In [42], the kernel method on the min-rank problem was proposed. In our context, because we have n matrices of size $n \times n$, this method works as follows. Let $A \in \mathcal{A}$ be of rank $\le r$. Suppose $v \in \mathbb{F}_q^n$ is in the kernel of A. Then construct $(\sum_{i \in [n]} x_i A_i) v = \mathbf{0}$, which consists of n linear equations in the variables $x_1, \dots, x_n$. It can be expected that the solution space is of 1-dimensional, and any non-zero solution gives rise to λA for some nonzero $\lambda \in \mathbb{F}_q$. So the problem is to sample a non-zero vector in $\ker(A)$. Since $|\ker(A)| = q^{n-r}$ we expect to get one after q^r many samplings.

To adapt the above to our setting incurs an extra cost. Recall the matrices B_i's in Definition 11, constructed from an alternating trilinear form $\phi \in \Lambda(n, q)$. Let $\mathcal{B}$ be the space of matrices spanned by B_i's. Suppose for some $B = \sum_{i \in [n]} u_i B_i \in \mathcal{B}$, we have $\mathrm{rk}(B) = r$. For $v \in \ker(B)$, set up the equations $(\sum_{i \in [n]} x_i B_i) v = \mathbf{0}$, and let S be the solution space. So S contains $u = (u_1, \dots, u_n)$ as promised, but S also contains v due to the alternating property. In other words, $\dim(S)$ is at least 2, and we need to go through all the lines in $\dim(S)$. This adds a multiplicative factor of q to the original sampling.

To use the above in solving ATFE, we can combine the above with the birthday paradox idea from [16] as follows. Let $\phi, \psi \in \mathrm{ATF}(n, q)$ be the input to ATFE. Suppose $\min\{\mathrm{rk}(\phi_u) : u \in \mathbb{F}_q^n\} = \min\{\mathrm{rk}(\psi_u) : u \in \mathbb{F}_q^n\}$, which should holds with high probability. Let r be this minimum rank and it is known to us (by carrying out experiments from Table 3). Suppose $R_{\phi,r}$ and $R_{\psi,r}$ (defined in Section 5.1) are of size $\approx s$, which can also be determined before hand using experiments. By birthday paradox, to create a collision it is enough to sampling $\sqrt{s}$ many $u \in \mathbb{F}_q^n$ from $R_{\phi,r}$ and $R_{\psi,r}$, respectively. Each sampling requires $q^r \cdot q$ many operations, so total sampling needs $q^{r+1} \cdot \sqrt{s}$ many operations. After that we get a collision (i.e. u and v such that $Au = v$ for some A sending ϕ to ψ) with constant probability by trying over $\sqrt{s} \cdot \sqrt{s} = s$ many possibilities. Because of Assumption 1, the rest can be solved in polynomial time.

From the above we see that the main multiplicative factors are $q^{r+1} \cdot \sqrt{s}$ and s. From Table 3 and Remark 3, when $n = 9$, we seem to have $r = 4$ and $s = q^3$ in this case. So $q^{r+1} \cdot \sqrt{s}$ becomes $q^{6.5} > q^{2/3 \cdot n}$. Combining with Assumption 1, the running time of the algorithm is not competitive with $O\big(q^{2/3 \cdot n} \cdot n^{2 \cdot \omega} \cdot \log_2(q)\big)$.

Remark 5. The Grover search should be helpful in the above setting. Note that it is easy to formulate searching a (non-zero) matrix of minimum rank as a marked search problem. Using the terminologies above, the multi-target Grover search can return a matrix of rank r (minimum rank) in quantum time $\sqrt{\frac{q^n}{s}}$. Therefore getting $\sqrt{s}$-many min-rank matrices requires quantum time $q^{n/2}$. This is better than q^{r+1} when $n \leq 2(r+1)$.

We also studied attacks based on the Kipnis-Shamir relinearisation attack [55] and that with the Gröbner basis method [36,34,35] in Appendix B.

6 Implementation results

6.1 Parameter Choices

The following analysis on parameters with respect to a fixed security level is modelled after CSI-FiSh [11]. We present here the choice of parameters for the basic scheme in Section 3.1. The choice of parameters for the Merkle variant is presented in Appendix A.

To achieve the security level as λ bits, there are four parameters to determine: n and q as in $\mathrm{ATF}(n, q)$, the round number r, and the number of alternating trilinear forms generated in each round which is $C = 2^c$. Some key criteria are as follows.

- First, by Equation (1) from the algorithmic study of ATFE, we use
$$\frac{2}{3} \cdot n \cdot \log_2(q) + 2\omega \cdot \log_2(n) + \log_2(\log_2(q)) \geq \lambda. \tag{5}$$
to estimate the bit complexity for solving ATFE. Here ω is the matrix multiplication exponent, and we take $\omega = 2$ for the sake of added security.
- As discussed in [11], as customary it is reasonable to assume that the probability of a successful attack is at most $Q \times E$, where Q is the number of hash function evaluations and E is the soundness error of the zero-knowledge Goldreich-Micali-Wigderson protocol. Therefore we require $C^{-r} \leq 2^{-\lambda}$, leading to
$$r \cdot c \geq \lambda. \tag{6}$$

It is possible to improve this a little bit by “slowing down” the hash function as indicated in [11], but we shall not explore this here.

- The scheme implies that the public key, private key, and signature sizes in terms of bytes are as follows.

$$\begin{aligned} \text{PubKeySize} &= 2^c \cdot \binom{n}{3} \cdot \lceil \log_2(q) \rceil / 8, && (7) \\ \text{PriKeySize} &= 2^c \cdot n^2 \cdot \lceil \log_2(q) \rceil / 8, && (8) \\ \text{SigSize} &= r \cdot (c + n^2 \cdot \lceil \log_2(q) \rceil)/8. && (9) \end{aligned}$$

We list our reasons for parameter choices.

The choice of n. As the public key size grows in n^3, we wish to keep n as small as possible. By the attack methods described in Section 5, the only attack method that is insensitive to q is the Gröbner basis attack. So we need to set up n such that the Gröbner basis attack fails.

The practical experiments with Gröbner basis as reported in Section 5.2 indicate setting $n \geq 9$. This is partly due to an important distinction between $n = 8$ and $n = 9$. That is, when $n = 8$, the dimension of $\mathrm{ATF}(8, q)$ (56) is less than the dimension of $\mathrm{GL}(8, q)$ (64). This weakness disappears at $n = 9$, when the dimension of $\mathrm{ATF}(9, q)$ (84) exceeds the dimension of $\mathrm{GL}(9, q)$ (81). Because of these, we choose n to be 9, 10, and 11 as first test beds.

The choice of q. Since the only thing that matters in our analysis about q is its size, it seems reasonable to set q to be a prime for simpler and hopefully faster implementation.

The choice of r and c. It seems reasonable to set r and $C = 2^c$ to be roughly the same. This is because C controls the public key size and r controls the signature size. Making r and C close helps to achieve the best performance regarding the sum of public key and signature sizes. We also need to ensure that $q^{2/3} \leq n^{12}$ so that $q^{2/3n} \leq n^{12n}$ by the Gröbner basis method analysis.

Based on the above, we provide two sets of parameter choices for the basic scheme as follows. Note that we include both $n = 8$ and $n = 9$, as odd and even numbers seem to demonstrate different behaviours regarding the rank statistics (Table 3), which in turn affects the attack methods.

6.2 A prototype implementation

In this section, we provide a prototype implementation of the basic scheme using C to evaluate its practical efficiency. We first explain some optimizations we have performed on our prototype. We then propose some concrete parameter sets and report the running times of our implementation.

1. First, to efficiently generate random invertible matrix A_i over $\mathbb{F}_q$, we generate each time two random matrices $L_i \in \mathbb{F}_q^{n,n}$ and $U_i \in \mathbb{F}_q^{n,n}$ such that

  - $L_{i,j} \xleftarrow{\$} \mathbb{F}_q$ if $i > j, L_{i,j} = 0$ if $i < j$ and $L_{i,i} = 1$,
  - $U_{i,j} \xleftarrow{\$} \mathbb{F}_q$ if $i < j, U_{i,j} = 0$ if $i > j$ and $U_{i,i} \xleftarrow{\$} \mathbb{F}_q^*$.

   Consequently, we can efficiently compute $A_i = L_i U_i$. The LU decomposition guarantees a bijection between the set of the invertible matrices in $\mathbb{F}_q$ and the $q^{n(n-1)}(q-1)^n$ possible combinations of L_i and U_i. It allows us to avoid determinant computation as well as re-sampling in case A_i is a singular matrix.
2. Secondly, we use a dedicated technique to perform modular reduction. Operating on matrices and tensors require multiple computations of a sum of products of elements over $\mathbb{F}_q$. To perform those inner products efficiently, we will start by computing the sum of products in $\mathbb{Z}$ and then performing a modular reduction, which we employ the algorithm proposed recently by Plantard in [73]. It allows us to efficiently reduce the resulting inner product, which is smaller than nq^2, to a number smaller than q by in just 2 multiplications, 2 bit-shifts and one addition; see [73] for more detail. In such situation, this choice is outperforming the usage of Pseudo-Mersenne number [26] as well as Montgomery reduction [66]. However, it requires two specific corrections:
   - the result is $\leq q$ not $< q$. This slight redundancy needs to be corrected at the final stage of computations; and
   - the modular arithmetic uses a Montgomery-like representation, i.e., in our computations, $x \bmod q$ will be represented by $x \cdot (-2^{64}) \bmod q$. However, the computation output do not need to be in such representation. Furthermore, to pick random elements in $\mathbb{F}_q$ or $\mathbb{F}_q^*$, one can assume those elements being already in such representation. Nevertheless, when initializing the identity matrix we will simply initiate 1 in the diagonal directly to $-2^{64} \bmod q$.

Our experience suggests that the main overhead lies at the computation of the group action. The procedure described in Section 2.1 requires $O(n^6)$ many operations. To bring this down we can first turn alternating trilinear forms into 3-way arrays as in Definition 11, and then perform $3n$ many matrix multiplications. This results in a procedure with $O(n^4)$ many operations. The alternating structure helps to reduce the constant hidden in the big O notation.

Remark 6. For modular arithmetics, we also implemented the Montgomery classic method [66], the reduction modulo a Pseudo-Mersenne method [26], as well as Seiler's method [76]. Our experiments suggested that, while there were no significant differences among these four methods, the method we use in this paper following [73] is the most efficient, as shown in Table 4.

Some concrete proposals. Based on the above analysis, we propose some concrete instantiations in Tables 5. For timing, we performed 10^5 tests for each function and computed the average. Our tests were performed on a Linux 5.11.0-37-generic with Intel Core i7-8565U CPU (1.80 GHz) and compiled with g++10.3.0 with "-Ofast -flto -fwhole-program -march=native" options.

	Time in μs		
Method	Set-Up	Sign	Verify
[73]	383.1	660.0	578.9
[66]	414.1	678.8	616.1
[26]	412.8	682.8	598.8
[76]	425.0	692.7	609.1

Table 4. The timings for four modular arithmetic methods for Option 2 in Table 5. The results were averaged over 10^5 tests for each method.

	Parameters					Size in Byte			Time in μs		
	n	q	r	c	λ	Public key	Private key	Signature	Set-Up	Sign	Verify
Option 1	9	524287	26	5	128	6384	6156	5018	285.9	471.7	416.5
Option 2	10	131071	26	5	128	8160	6800	5542	383.1	660.0	578.9
Option 3	10	131071	32	4	128	4080	3400	6816	190.7	795.4	708.8
Option 4	11	65521	26	5	128	10560	7744	6309	514.0	861.1	765.2

Table 5. Proposed concrete instantiations of the proposed scheme.

Acknowledgement.

We thank the reviewers for their careful reading and several questions and suggestions. We thank Luca De Feo for his careful reading and pointing out some typos. Y.Q. was partly supported by the Australian Research Council Discovery Projects DP200100950. D.H.D. and W.S. were partly supported by the Australian Research Council Linkage Projects LP190100984. A.J. was supported by the European Union's H2020 Programme under grant agreement number ERC-669891.

References

1. M. Agrawal and N. Saxena. Automorphisms of finite rings and applications to complexity of problems. In *STACS 2005*, pages 1–17, 2005.
2. M. Agrawal and N. Saxena. Equivalence of f-algebras and cubic forms. In *STACS 2006*, pages 115–126, 2006.
3. G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, J. Kelsey, Y.-K. Liu, C. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, and D. Smith-Tone. Status report on the second round of the NIST post-quantum cryptography standardization process. Technical report, National Institute of Standards and Technology, July 2020.
4. N. Alamati, L. D. Feo, H. Montgomery, and S. Patranabis. Cryptographic group actions and applications. In *Advances in Cryptology - ASIACRYPT 2020*, pages 411–439. Springer, 2020.
5. MD Atkinson. Alternating trilinear forms and groups of exponent 6. *Journal of the Australian Mathematical Society*, 16(1):111–128, 1973.
6. L. Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *STOC 2016*, pages 684–697, 2016.

7. S. Bai, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. Crystals-dilithium: Algorithm specifications and supporting documentation (version 3.1). `https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf`, 2021.
8. M. Bardet. *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2004.
9. M. Bardet, J.-C. Faugère, B. Salvy, and B.-Y. Yang. Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. In *Proc. of MEGA*, volume 5, 2005.
10. J. Berthomieu, J.-C. Faugère, and L. Perret. Polynomial-time algorithms for quadratic isomorphism of polynomials: The regular case. *J. Complexity*, 31(4):590–616, 2015.
11. W. Beullens, T. Kleinjung, and F. Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In *ASIACRYPT 2019*, pages 227–247. Springer, 2019.
12. M. Bellare, G. Neven. Multi-signatures in the plain public-Key model and a general forking lemma. In *CCS 2006*, pages 390–399, 2016.
13. X. Bonnetain and A. Schrottenloher. Quantum security analysis of CSIDH. In *EUROCRYPT 2020* , pages 493–522. Springer, 2020.
14. C. Bouillaguet. *Etudes d'hypotheses algorithmiques et attaques de primitives cryptographiques*. PhD thesis, PhD thesis, Université Paris-Diderot–École Normale Supérieure, 2011.
15. 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 *PKC 2011*, pages 473–493. Springer, 2011.
16. C. Bouillaguet, P.-A. Fouque, and A. Véber. Graph-theoretic algorithms for the "isomorphism of polynomials" problem. In *EUROCRYPT 2013*, pages 211–227, 2013.
17. G. Brassard and M. Yung. One-way group actions. In *CRYPTO 1990*, pages 94–107, 1990.
18. P. A. Brooksbank, Y. Li, Y. Qiao, and J. B. Wilson. Improved algorithms for alternating matrix space isometry: from theory to practice. In *28th ESA 2020*, 26:1–26:15, 2020.
19. P. A. Brooksbank, J. Maglione, and J. B. Wilson. A fast isomorphism test for groups whose Lie algebra has genus 2. *J. Algebra*, 473:545–590, 2017.
20. J. F Buss, G. S Frandsen, and J. O Shallit. The computational complexity of some problems of linear algebra. *Journal of Computer and System Sciences*, 58(3):572–596, 1999.
21. N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *EUROCRYPT 2000*, pages 392–407. Springer, 2000.
22. W. Castryck, T. Lange, C. Martindale, Lorenz Panny, and J. Renes. CSIDH: an efficient post-quantum commutative group action. In *ASIACRYPT 2018*, pages 395–427. Springer, 2018.
23. A. Childs, D. Jao, and V. Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014.
24. A. M. Cohen and A. G. Helminck. Trilinear alternating forms on a vector space of dimension 7. *Communications in algebra*, 16(1):1–25, 1988.

25. J. M. Couveignes. Hard homogeneous spaces. *IACR Cryptology ePrint Archive*, 2006.
26. R. E. Crandall Method and apparatus for public key exchange in a cryptographic system. *U.S. Patent number 5159632*, 1992.
27. L. D. Feo, D. Kohel, A. Leroux, C. Petit, and B. Wesolowski. Sqisign: compact post-quantum signatures from quaternions and isogenies. In *ASIACRYPT 2020*, pages 64–93. Springer, 2020.
28. J. Ding and D. Schmidt. Rainbow, a new multivariable polynomial signature scheme. In *ACNS 2005*, pages 164–175. Springer, 2005.
29. J. Don, S. Fehr, C. Majenz, and C. Schaffner. Security of the fiat-shamir transformation in the quantum random-oracle model. In *CRYPTO 2019*, pages 356–383. Springer, 2019.
30. J. Draisma and R. Shaw. Some noteworthy alternating trilinear forms. *Journal of Geometry*, 105(1):167–176, 2014.
31. A. El Kaafarani, S. Katsumata, and F. Pintore. Lossy csi-fish: Efficient signature scheme with tight reduction to decisional CSIDH-512. In *PKC 2020*, pages 157–186. Springer, 2020.
32. P. Erdős and A. Rényi. Asymmetric graphs. *Acta Mathematica Hungarica*, 14(3-4):295–315, 1963.
33. J.-C. Faugère. A new efficient algorithm for computing gröbner bases without reduction to zero (F5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, pages 75–83, 2002.
34. J.-C. Faugere, M. S. El Din, and P.-J. Spaenlehauer. Computing loci of rank defects of linear matrices using gröbner bases and applications to cryptology. In *ISSAC 2010*, pages 257–264, 2010.
35. J.-C. Faugere, M. S. El Din, and P.-J. Spaenlehauer. On the complexity of the generalized minrank problem. *Journal of Symbolic Computation*, 55:30–58, 2013.
36. J.-C. Faugere, F. Levy-dit Vehel, and L. Perret. Cryptanalysis of minrank. In *CRYPTO 2008*, pages 280–296. Springer, 2008.
37. J.-C. Faugère and L. Perret. Polynomial equivalence problems: Algorithmic and theoretical aspects. In *EUROCRYPT 2006*, pages 30–47, 2006.
38. L. D. Feo and S. D. Galbraith. Seasign: Compact isogeny signatures from class group actions. In *EUROCRYPT 2019*, pages 759–789. Springer, 2019.
39. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO 1986*, pages 186–194, 1986.
40. P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. Falcon: Fast-fourier lattice-based compact signatures over ntru (specification v1.2). `https://falcon-sign.info/falcon.pdf`, 2020.
41. O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.
42. Lo. Goubin and N. T. Courtois. Cryptanalysis of the TTM cryptosystem. In *ASIACRYPT 2000*, pages 44–57. Springer, 2000.
43. M. Grigni, L. J. Schulman, M. Vazirani, and U. V. Vazirani. Quantum mechanical algorithms for the nonabelian hidden subgroup problem. *Comb.*, 24(1):137–154, 2004.
44. J. A. Grochow and Y. Qiao. On the complexity of isomorphism problems for tensors, groups, and polynomials I: tensor isomorphism-completeness. In *ITCS 2021*, pages 31:1–31:19, 2021.

45. J. A. Grochow and Y. Qiao. On p-Group Isomorphism: Search-To-Decision, Counting-To-Decision, and Nilpotency Class Reductions via Tensors. In *CCC 2021*, pages 16:1–16:38, 2021.
46. J. A. Grochow, Y. Qiao, and G. Tang. Average-case algorithms for testing isomorphism of polynomials, algebras, and multilinear forms. In *STACS 2021*, 38:1–38:17, 2021.
47. L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.
48. S. Hallgren, C. Moore, M. Rötteler, A. Russell, and P. Sen. Limitations of quantum coset states for graph isomorphism. *J. ACM*, 57(6):34:1–34:33, November 2010.
49. J. Håstad. Tensor rank is NP-complete. *J. Algorithms*, 11(4):644–654, 1990.
50. J. Hora and P. Pudlák. Classification of 8-dimensional trilinear alternating forms over gf (2). *Communications in Algebra*, 43(8):3459–3471, 2015.
51. G. Ivanyos and Y. Qiao. Algorithms based on *-algebras, and their applications to isomorphism of polynomials with one secret, group isomorphism, and polynomial identity testing. *SIAM Journal on Computing*, 48(3):926–963, 2019.
52. W. Beullens, C. Dobraunig, M. Eichlseder, S. Fluhrer, S.-L. Gazdag, A. Hülsing, P. Kampanakis, S. KÏbl, T. Lange, M. M. Lauridsen, F. Mendel, R. Niederhagen, C. Rechberger, J. Rijneveld, P. Schwabe. B. Westerbaan, J.-P. Aumasson, D. J. Bernstein. Sphincs+: Submission to the nist post-quantum project, v.3. `https://sphincs.org/data/sphincs+-round3-specification.pdf`, 2020.
53. Z. Ji, Y. Qiao, F. Song, and A. Yun. General linear group action on tensors: A candidate for post-quantum cryptography. In *TCC 2019*, pages 251–281. Springer, 2019.
54. D. Jao and L. De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *PQCrypto 2011*, pages 19–34.
55. A. Kipnis and A. Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In *CRYPTO 1999*, pages 19–30. Springer, 1999.
56. J. Köbler, U. Schöning, and J. Torán. *The Graph Isomorphism Problem*. Basel Birkhüser, 1993.
57. G. Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM J. Comput.*, 35(1):170–188, 2005.
58. G. Kuperberg. Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. In *TQC 2013*, pages 20–34, 2013.
59. Y. Li and Y. Qiao. Linear algebraic analogues of the graph isomorphism problem and the Erdős–Rényi model. In *FOCS 2017*, pages 463–474. IEEE Computer Society, 2017.
60. Q. Liu and M. Zhandry. Revisiting post-quantum fiat-shamir. In *CRYPTO 2019*, pages 326–355. Springer, 2019.
61. Waterloo Ontario. Maplesoft, a division of Waterloo Maple Inc. Maple (2020.2), 2020.
62. B. D. McKay. Practical graph isomorphism. *Congr. Numer.*, pages 45–87, 1980.
63. B. D. McKay and A. Piperno. Practical graph isomorphism, II. *J. Symb. Comput.*, 60:94–112, 2014.
64. R. C. Merkle. A certified digital signature. In *CRYPTO 1989*, pages 218–238. Springer, 1989.
65. N. Midoune and L. Noui. Trilinear alternating forms on a vector space of dimension 8 over a finite field. *Linear and Multilinear Algebra*, 61(1):15–21, 2013.
66. P. L. Montgomery. Modular Multiplication Without Trial Division. *Mathematics of Computation*, 44:519–521, 1985.

67. C. Moore, A. Russell, and L. J. Schulman. The symmetric group defies strong fourier sampling. *SIAM J. Comput.*, 37(6):1842–1864, 2008.
68. C. Moore, A. Russell, and U. Vazirani. A classical one-way function to confound quantum adversaries. *arXiv preprint quant-ph/0701115*, 2007.
69. D. Moody. The Homestretch: the beginning of the end of the NIST PQC 3rd Round, PQCrypto 2021. Available from `https://pqcrypto2021.kr/download/program/2.2_PQCrypto2021.pdf`
70. E. A. O'Brien. Isomorphism testing for p-groups. *Journal of Symbolic Computation*, 17(2):133–147, 1994.
71. J. Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In *EUROCRYPT 1996*, pages 33–48, 1996.
72. C. Peikert. He gives c-sieves on the CSIDH. In *EUROCRYPT 2020*, pages 463–492. Springer, 2020.
73. T. Plantard. Efficient Word Size Modular Arithmetic. *IEEE Transactions on Emerging Topics in Computing*, 9(3):1506–1518,2021.
74. O. Regev. Quantum computation and lattice problems. *SIAM J. Comput.*, 33(3):738–760, 2004.
75. L. J. Schulman. Cryptography from tensor problems. *IACR Cryptol. ePrint Arch.*, 2012:244, 2012.
76. G. Seiler. Faster AVX2 optimized NTT multiplication for Ring-LWE lattice cryptography. *IACR Cryptol. ePrint Arch.*, 2018:039, 2018.
77. N. Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Trans. Information Theory*, 46(4):1193–1203, 2000.
78. M. S. Chen, J. Ding, M. Kannwischer, J. Patarin, A. Petzoldt, D. Schmidt, and B-Y. Yang. Rainbow signature: One of the three nist post-quantum signature finalists. `https://www.pqcrainbow.org/`, 2021.
79. P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.
80. A. Stolbunov. *Cryptographic schemes based on isogenies.* PhD thesis, Norwegian University of Science and Technology, 2012.
81. J. B. Wilson. Decomposing p-groups via Jordan algebras. *Journal of Algebra*, 322(8):2642–2679, 2009.
82. E. M. Wright. Graphs on unlabelled nodes with a given number of edges. *Acta Mathematica*, 126(1):1–9, 1971.
83. J. Y. C. Yeh, C. M. Cheng, B. Y. Yang. Operating Degrees for XL vs. F4/F5 for Generic MQ with Number of Equations Linear in That of Variables. *Number Theory and Cryptography 2013*, 19–33.

A The Merkle tree variant

Merkle trees [64] are used widely in signature scheme designs, including recent works on isogeny-based signatures [38,11]. Our use is completely analogous to that in [11].

Let us first give a sketch here. Recall that the public key in the basic scheme is $(\phi_1, \dots, \phi_C)$, $\phi_i \in \mathrm{ATF}(n, q)$, where $C = 2^c$. Let h be a hash function. Build a complete binary tree T of depth c. For each node v we assign (a, b), where $a \in \{0, 1, \dots, c\}$ denotes the depth v is on, and $b \in [2^a]$. In particular, the children of (a, b) are $(a+1, 2b-1)$ and $(a+1, 2b)$. Then label the leaf node (c, i) with $\ell_{c,i} := h(\phi_i | C + i | \text{MerkleKey})$, where MerkleKey $\in \{0,1\}^\lambda)$ is a random bit string, included in both public and private keys. Each internal node (a, b) is labelled by $\ell_{a,b} := h(\ell_{a+1,2b-1} | \ell_{a+1,2b} | 2^a + b | \text{MerkleKey})$. The public key is then the label of the root and MerkleKey. The private key consists of MerkleKey, ϕ_C, and the original matrices $A_1, \dots, A_C \in \mathrm{GL}(n, q)$. The signing procedure is the same except that now we need to also include, for $i \in [r]$, ϕ_{b_i}, and the authentication path of ϕ_{b_i} on the Merkle tree, which consists of hash values of those nodes adjacent to the path from (c, b_i) to the root. The verification procedure starts by verifying that ϕ_{b_i}'s are on the tree, and the rest is the same.

It is possible to reduce the private key size to say 32 bytes, by using a pseudorandom generator to generate A_i and ϕ_C.

For the choice of parameters of the Merkle tree variant, Equations (5) and (6) still apply. For the other parameters, we have the following (in terms of bytes), assuming we use a hash function producing 256-bit outputs.

$$\begin{aligned} \text{PubKeySize} &= & 48, \quad (10)\\ \text{PriKeySize} &= & (2^c \cdot n^2 \cdot \lceil \log_2(q) \rceil + \binom{n}{3} \cdot \lceil \log_2(q) \rceil + 128)/8, \quad (11)\\ \text{SigSize} &= & r \cdot (c + n^2 \cdot \lceil \log_2(q) \rceil + \binom{n}{3} \cdot \lceil \log_2(q) \rceil + 256 \cdot c)/8. \quad (12) \end{aligned}$$

Note that for public key and private sizes, 16 bytes are needed for MerkleKey.

Therefore, the choices of n and q are the same as the basic scheme. For r and c, it is desirable to set r as small as possible, because it is key in the signature size. However, note that r cannot be set too small, as this implies large c which results in a large key generation time.

B On the Kipnis-Shamir relinearisation attack

The Kipnis-Shamir relinearisation attack. Suppose we wish to solve a system of m quadratic equations in n variables $x_1, \dots, x_n$. When $m \approx n^2/2$, the linearisation method assigns new variables $y_{i,j}$ for quadratic monomials $x_i x_j$, solves the resulting linear equations, and recovers a solution to x_i's from the solution to linear equations. In [55], Kipnis and Shamir develops the relinearisation method which allows to work in the setting when $m = \epsilon \cdot n^2$ for small constant ϵ. Furthermore the relinearisation helps to relieve the problem that many solutions to linear equations do not correspond to the original solutions to quadratic equations.

An application of the Kipnis-Shamir relinearisation to the min-rank problem is to compute rank-1 matrices in the matrix space, as one can write out the 2×2 minors and solve the system of quadratic equations; see [75] for a concrete use. This does not apply to our setting because by Table 3, a random alternating trilinear form $\phi \in \mathrm{ATF}(n, q)$ do not yield ϕ_u of rank ≤ 3 when $n \geq 8$.

There are two ways to extend the above idea to rank-r matrices.

The first approach is straightforward: write out the $(r+1) \times (r+1)$ minors, assign a new variable to each degree-$(r+1)$ term, and solve the linear equations. However, if there are many rank-r matrices (just as in our case; cf. Table 3), the solution space could be of large dimension, so many solutions do not correspond to original solutions. To address this issue one needs to add in more relations among these newly introduced variables (the *re*linearisation idea), but this quickly becomes messy so not easy to understand or analyse.

The second approach was to combine the Kipnis-Shamir method with the Gröbner basis method. This was introduced by Faugère, Levy-dit-Vehel and Perret in [36] and further developed in [34,35]. This will be explained next.

The Kipnis-Shamir and Gröbner basis attack. In [36,34,35] the Kipnis-Shamir method was rephrased and combined with the Gröbner basis method. When the number of matrices is $(n-r)^2$, the algorithm there runs in time $O(n^{\omega \cdot (n-r)^2})$. When $n-r=3$, experiments also suggest the efficiency in practice.

It is unclear whether these results apply to our setting, as from Definition 11, we have n (instead of $(n-r)^2$) matrices B_i's of size $n \times n$ with certain structural restrictions. It is certainly of value to investigate this in the future.