Trilinear Form Equivalence Problems: from Algorithm and Complexity to Post-Quantum Cryptography.

by

Gang Tang

A thesis submitted in satisfaction of the

requirements for the degree of

Doctor of Philosophy

in the

Faculty of Engineering and Information Technology

at the

University of Technology Sydney

Supervisor: Youming Qiao

Co-supervisor: Mingsheng Ying

March 2024

Certificate of Original Authorship

I, Gang Tang, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior to publication.

Signature of Author

Date: 02-Mar-2024

Abstract

Trilinear Form Equivalence Problems: from Algorithm and Complexity to Post-Quantum Cryptography.

by

Gang Tang

Doctor of Philosophy

University of Technology Sydney

In this thesis, we present new results on alternating trilinear form equivalence (ATFE), a problem that arises in both mathematics and cryptography. ATFE is of particular interest due to its connections to various hard problems in cryptography, which makes it a promising candidate for post-quantum cryptographic schemes. We study its complexity, cryptographic applications and algorithms. We also study the QROM security and the construction of linkable ring signatures based on group action under the GMW-FS framework.

- **Digital signature from ATFE.** We study the complexity of testing equivalence of alternating trilinear forms. This problem is of interest in both mathematics and cryptography. We show that this problem is polynomial-time equivalent to testing equivalence of symmetric trilinear forms, by showing that they are both Tensor Isomorphism-complete, therefore is equivalent to testing isomorphism of cubic forms over most fields.

  We then propose a post-quantum signature scheme ALTEQ based on the ATFE problem. We implement the ALTEQ scheme with several optimizations and submit it to NIST call for additional post-quantum signature standardization as a round 1 candidate.
- **GMW-FS design based on group actions.** Group action based cryptography was formally proposed in the seminal paper of Brassard and Yung (Crypto

1990). Based on one-way group action, there is a well-known digital signature design based on the Goldreich–Micali–Widgerson (GMW) zero-knowledge protocol for the graph isomorphism problem and the Fiat–Shamir (FS) transformation.

Our second result concerns the QROM security and ring signatures of the GMW-FS design. We distil properties of the underlying group action for the GMW-FS design to be secure in the quantum random oracle model (QROM). We also show that this design supports a (linkable) ring signature construction following the work of Beullens, Katsumata and Pintore (Asiacrypt 2020). We apply these results to support the security of the ALTEQ scheme in the QROM model. We then describe a linkable ring signature scheme based on it, and provide an implementation of the ring signature scheme. Preliminary experiments suggest that our scheme is competitive among existing post-quantum ring signatures.

- **Cryptanalysis on MEDS and ALTEQ.** The final part of this thesis investigates the hardness of MCE and ATFE. We present new algorithms for MCE and ATFE, which are the further development of the algorithms for polynomial isomorphism and alternating trilinear form equivalence, in particular by Bouillaguet, Fouque, and Véber (Eurocrypt 2013), and Beullens (Crypto 2023). Key ingredients in these algorithms are new easy-to-compute distinguishing invariants under the respective group actions. This algorithm has some implications for the security of MEDS. our algorithm for ATFE improves on the runtime of the previously best known algorithm of Beullens (Crypto 2023).

Acknowledgements

Time flies indeed, and the four years of PhD are so short. Met so many interesting things and interesting people here. Every experience I've had in Sydney has become a cherished memory, a tapestry of memories I will always cherish.

First of all, my heartfelt gratitude goes to my supervisor, Youming Qiao. From the very beginning, when I embarked on this journey knowing little—I was starting from nothing. Youming was very generous and devoted a lot of time to giving me very careful guidance (I still miss the regular meetings with you twice a week). As I gradually found my footing, Youming gracefully transitioned into nurturing my independence, shaping me into a researcher poised for long-term career development. I would also to thank Youming for supporting my participation in numerous workshops and conferences, each a stepping stone in my academic growth.

Next, I would like to thank Mingsheng Ying and Sanjiang Li, without their help I would not have the opportunity to start my PhD here. I am grateful to Sanjiang Li for organizing the quantum computing and information reading group.

I also want to thank all the collaborators: Markus Bläser, Zhili Chen, Dung Hoang Duong, Joshua A Grochow, Antoine Joux, Anand Kumar Narayanan, Ngoc Tuong Nguyen, Thomas Plantard, Arnaud Sipasseuth, Willy Susilo and Chuanqi Zhang. Without your invaluable contributions, this thesis would not have been possible. I look forward to working with you again in the future.

I thank all the members in CQSI for making it such a nice place to work. In particular, I thank Troy Lee for his help with my candidate accessment 1,2 and 3. I am grateful to Michael Bremner for supporting me in traveling for conferences. I thank Robyn Barden, Camila Cremonese, Margot Kopel, Esin Morcos, and Lily Qian for their help with various procedures.

Outside of academia, I want to thank my family and friends. My parents, in particular, have been steadfast pillars of support and encouragement throughout this jour-

ney. And to my girlfriend, Xiaojin Yang, your unwavering emotional support has been a beacon of light during moments of frustration and challenge.

To each and every one of you, thank you.

Contents

Chapter 1

Introduction

Public-key cryptosystems widely used today rely on hard problems like integer factorization and discrete logarithm problems over elliptic curves or finite fields. However, Shor's algorithms [Sho97], proposed in 1997, solve these problems in polynomial time on a quantum machine. Despite the current absence of large-scale and universal quantum computers, the rapid advancements in quantum technology in recent years have raised concerns in the cryptography community. While the exact timeline of this threat remains uncertain, the imperative to develop new cryptographic systems resistant to quantum attacks is pressing, leading to the emergence of *post-quantum cryptography*.

The development of post-quantum cryptography has inspired researchers across various fields to explore novel mathematical problems that are hard for quantum computers. These problems can be broadly categorized into five groups: lattice-based, code-based, multivariate-based, hash-based, and isogeny-based. We give an overview of these five categories in Section 1.1.

In 2016, the National Institute of Standards and Technology (NIST) launched the standardization for post-quantum cryptosystems, aiming to proactively migrate to post-quantum cryptosystems to deal with potential threats. This initiative has significantly piqued academic and industrial interest in practical post-quantum cryptosys-

tems. In 2022, four cryptosystems were selected for standardization: the lattice-based encryption scheme KYBER [SAB+22], the lattice-based signature schemes Dilithium [BDK+21] and Falcon [FHK+20], and the hash-based signature scheme SPHINCS+ [ABWB+20]. Additionally, three code-based and one isogeny-based scheme, namely Classical McEliece [BCC+22], HQC [MAB+22], BIKE [ABB+22] and SIKE [JAC+22] [1], moved to the final round of standardization for further cryptanalysis.

Embracing the principle of diversity for standardization, NIST [oST22] launched a new call for post-quantum signature standardization in 2022, with a particular emphasis on the cryptosystems based on a variety of different problems. Notably, the ALTEQ [BDN+23] signature, introduced in the Chapter 3 of this thesis, has been submitted to NIST standardization as the first-round candidate. We provide a comparison between ALTEQ and some NIST submissions, please refer to Section 3.5.4.

1.1 Overview of post-quantum cryptography.

In this section, we give an overview of the main families of post-quantum cryptography.

Lattice-based cryptography

Lattice-based cryptographic primitives are widely regarded as one of the most promising post-quantum cryptographic approaches, extensively explored both theoretically and practically. At the core of lattice cryptography lie two fundamental problems: Learning with Errors (LWE) and Short Integer Solutions (SIS). These problems form the cornerstone for constructing lattice-based cryptographic protocols. SIS, introduced as early as 1996 by Ajtai, marked a pivotal advancement, with Ajtai [Ajt96] providing a reduction from the worst-case approximate shortest vector problem to the average-case SIS problem. Another seminal contribution came from Regev [Reg05] in 2005, who introduced LWE and demonstrated reductions from some worst-case

[1] SIKE [JAC+22] suffered fatal attacks [CD23, MMP+23, Rob23] and should no longer be used

problems in lattice to average-case LWE. A notable advantage of LWE is its applicability in constructing public key encryption schemes and various other cryptosystems. For instance, the pioneering lattice-based public key encryption scheme was proposed by Regev in [Reg05]. To enhance the practicality of lattice-based cryptosystems (e.g. low size and high efficiency), several variants of LWE/SIS have been proposed, including ring-LWE/SIS and modular-LWE/SIS, all of which enjoy worst-case to average-case reductions. Building upon LWE/SIS and their variants, numerous advanced cryptographic systems have been developed, such as fully homomorphic encryption [Gen09], attribute-based encryption [GVW13] etc [GVW15, BVWW16]. As for the basic cryptographic systems such as public key encryption and digital signature, there are three schemes selected by NIST for standardization: Kyber [SAB+22], Dilithium [BDK+21], Falcon [FHK+20].

Code-based cryptography

Code-based cryptography has a long history, and its security relies on the hard problems in coding theory, e.g. Learning Parity with Noise (LPN) and Syndrome Decoding (SD). In 1978, Berlekamp, McEliece, and Van Tilborg [BMVT78] proved that decoding random linear codes is NP-complete. In the same year, McEliece [McE78] proposed the pioneering code-based public key encryption cryptosystem, where the private key is a random binary irreducible Goppa code and the public key is a random generator matrix of a randomly permuted version of that code. The ciphertext consists of the corrupted codeword, namely, the codeword combined with a random error. Efficiently recovering the codeword, a process called decoding, is achievable using the private key, thereby eliminating errors. The adversary's challenge lies in attempting to decode the ciphertext armed only with knowledge of the public key and ciphertext. Moreover, there are three schemes in the final round of NIST standardization: Classical McEliece [BCC+22], HQC [MAB+22] and BIKE [ABB+22].

Multivariate-based cryptography

Multivariate cryptography is cryptographic primitives based on multivariate polynomials over finite fields, with its security relies on solving systems of multivariate polynomials over these fields. Multivariable cryptography has an old history in PQC. In 1988, Matsumoto and Imai [MI88] introduced the first multivariate-based public key cryptosystem, known as MI. However, in 1995, MI was broken by the linearized equation attack proposed by Patarin [Pat95]. Subsequently, in 1996, Patarin [Pat96] generalized the field by introducing Hidden Field Equation (HFE), which spawned some signature schemes. Another important class of multivariate schemes is based on the unbalanced oil and vinegar (UOV) scheme, initially proposed by Kipnis et al. [KPG99]. UOV is the unbalanced variant of oil and vinegar (OV), the latter one was broken by Kipnis-Shamir attack [KS98]. In 2005, Ding et al. [DS05] proposed Rainbow, a layered generalization of UOV, which was selected for the third round of the NIST PQC standardization due to its good performance and security. Recently, Beullens proposed a new hybrid algebraic attack and an improvement of the rectangular MinRank attacks, so that Rainbow-I can be practically solved. In the recent NIST call for additional post-quantum signature, UOV [BCD+23] was re-proposed, and nine other protocols based on multivariate were shortlisted.

Hash-based cryptography

One of the important properties of hash functions is that it is hard to compute the pre-image for a given output. The signature scheme based on the hash function was first proposed by Lamport [Lam79] in 1975. However, the signature size of Lamport's scheme is very large, and it is very demanding that each key can only sign one message. In this way, this scheme becomes an example of "one-time signature". To overcome the problems of large public keys that could be used only once, Merkle [Mer89] introduced an improved signature scheme in 1989. This signature scheme evolved from a one-time signature scheme, using the authentication mechanism of the Merkle

tree(hash tree). In a Merkle tree, the secret key in Lamport's scheme is used as the leaf node, and the root is the new public key in Merkle's scheme. The length of the public key can be effectively reduced by Merkle tree. The security of hash-based signature algorithms depends on the collision resistance of the hash function. Since there is no effective quantum algorithm that can quickly find the collision of the hash function, the signature scheme based on the hash function can effectively resist quantum computer attacks. In addition, the security of a hash-based digital signature scheme does not depend on a specific hash function. In other words, even if some of the currently used hash functions are breached, we can replace them with more secure hash functions. Further, there is one scheme selected by NIST for standardization: Sphincs+ [ABWB+20].

Isogeny-based cryptography

Isogeny-based cryptography is based on finding an isogeny map between elliptic curves. While relatively young in post-quantum cryptography, its roots trace back decades. In 2006, Couveignes [Cou06] and Rostovtsev and Stolbunov [RS06] independently introduced the first key exchange protocol based on isogenies. Its security is based on the hardness of finding isogeny on two given ordinary elliptic curves. The concept of supersingular elliptic curves was initially integrated into the CGL hash function from expander graphs, originating from Charles, Goren, and Lauter [CLG09]. In 2011, Jao and De Feo [JF11] proposed Supersingular Isogeny Diffie–Hellman (SIDH), a public key exchange protocol based on the isogeny between two supersingular elliptic curves. In practical applications, the SIKE [JAC+22] scheme, a variant of SIDH, moved as a final candidate for NIST standardization. However, due to recent advancements [CD23, MMP+23, Rob23] in the hardness of SIDH, SIKE is now considered vulnerable. Another notable isogeny-based key exchange protocol proposed in 2018 by Castryck, Lange, Martindale, Panny, and Renes [CLM+18], named Commutative Supersingular Isogeny Diffie–Hellman (CSIDH). CSIDH follows the framework laid

by [Cou06, RS06] but restricts its focus to supersingular elliptic curves defined over F_q. Despite it has lower computational efficiency compared to SIDH, CSIDH is the first post-quantum Non-Interactive Key Exchange (NIKE). Beyond these public key exchange protocols, many other isogeny-based primitives have also been developed, such as digital signatures [BKV19, FG19, DFKL+20], threshold signatures [FM20], ring and group signatures [BKP20, BDK+22]. It is noteworthy that in NIST's recent call for signature schemes, an isogeny-based scheme called SQISign [CSSF+23] has been proposed as the first-round candidate.

1.2 Overview of group action based cryptography

The use of group actions in cryptography has a long tradition. Indeed, the discrete logarithm problem can be interpreted as a problem about cyclic group actions [Cou06]. As far as we know, the first treatment of *abstract* group actions in cryptography goes back to Brassard and Yung [BY90], who proposed the notion of *one-way* group actions. When the groups are abelian (commutative), this was further developed by Couveignes [Cou06]. Recently, two independent works [JQSY19] and [AFMP20] enriched this framework further by introducing the notion of (weakly) pseudorandom group actions, which generalizes the celebrated Decisional Diffie–Hellman assumption [DH76, Bon98].

Besides setting up frameworks, many cryptographic primitives can be realized, such as claw-free one-way functions and bit commitment [BY90], quantum-secure pseudorandom functions [JQSY19] and zero-knowledge identification protocols [Cou06, JQSY19]. When the groups are abelian (commutative), more functions are possible, such as key exchange [Cou06, CLM+18], smooth projective hashing, and dual-mode public-key encryption [AFMP20].

Since discrete logarithms can be solved efficiently on quantum computers [Sho97], it is desirable to explore group actions suitable for post-quantum cryptography. Indeed, there are some instantiations of post-quantum security group actions.

These include lattice isomorphism [DvW22], Commutative Supersingular Isogeny Diffie–Hellman (CSIDH) [CLM+18], Alternating Trilinear Form Equivalence (ATFE) [TDJ+22], Linear Code Equivalence (LCE) [BMPS20], and Matrix Code equivalence (MCE) [CNP+23b]. These instantiations have been further developed into a variety of quantum-resistant cryptographic primitives, including digital signatures [TDJ+22, BMPS20, CNP+23b, BKV19, FG19], ring signatures [BKP20, BCD+22], and group signatures [BDK+22]. Many of the signature schemes use a unified framework, that is, the GMW-FS framework we described in Section 1.3.

As in the lattice case [Reg04], the research into hidden subgroup problems is of particular relevance here, especially the hidden shift problems [CJS14] and symmetric or general linear groups [HMR+10]. For the class group actions in the isogeny setting, even though the group action underlying CSIDH is commutative, the best quantum algorithms are still subexponential [Pei20, BS20]. For the group actions on LCE, ATFE and MCE, the groups are symmetric or general linear groups, so the previous negative evidence for standard techniques (such as coset sampling) in the hidden subgroup problem for graph isomorphisms [HMR+10] applies. 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 [Reg04], 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.

1.3 Overview of GMW-FS framework

In [GMW91], Goldreich, Micali and Wigderson described a zero-knowledge proof protocol for the graph isomorphism (GI) problem. We describe the *graph isomorphism* problem and this protocol as below.

Graph isomorphism problem. Given two graphs $G = ([n], E)$ and $H = ([n], F)$, where $[n]$ denotes $\{1, \ldots, n\}$ and E, F are the edge sets. The classical *graph isomorphism* problem asks whether two graphs are the same up to relabeling the vertices. Specifically, G and H are isomorphic if and only if there exists a permutation $\sigma : [n] \longrightarrow [n] (\sigma \in S_n)$ such that $\sigma(E) = F$, where $\sigma(E) = F$ means that for any $\{i, j\} \in E$ if and only if $\{\sigma(i), \sigma(j)\} \in F$.

GMW protocol for graph isomorphism. Given two graphs G and H as the statement, let σ be an isomorphism as the witness such that $\sigma(G) = H$. We then give the sigma protocol for graph isomorphism as follows:

(1) Commitment phase. The prover generates a random permutation π which sends G to $K = \pi(G)$ and then sends the new graph K to the verifier as the commitment,

(2) Challenge phase. The verifier random generates a binary challenge $b \in \{0, 1\}$ and sends it to the prover.

(3) Response phase. If $b = 0$ the prover sends $r := \pi$ to verifier; otherwise sends $r := \pi\sigma^{-1}$.

(4) Verification phase. If $b = 0$ the verifier accepts when $r(G) = K$; otherwise accepts when $r(H) = K$.

It's straightforward to check this protocol satisfies the completeness, special soundness and honest verifier zero knowledge.

The Fiat–Shamir transformation FS [FS86] can be applied to the above sigma protocol to yield a digital signature scheme. This construction has been observed by several researchers since the 1990's. However, this scheme based on the graph isomorphism is not secure, because GI can be solved efficiently in practice [McK80, MP14], not to mention Babai's quasipolynomial-time algorithm [Bab16]. 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 [Pat96] 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 3.3.8 of Section 3.3.2). Some problems, such as the isomorphism of quadratic polynomials with one secret, turn out to be easy [FP06, BFFP11, IQ19].

In isogeny-based cryptography, Couveignes [Cou06] first proposed the use of class group actions on elliptic curves in cryptography. He adapted the GMW identification protocol to this action. Stolbunov [Sto12] suggested to apply the Fiat-Shamir transformation to this identification protocol to get a signature scheme. However, the use of ordinary elliptic curves have issues including the subexponential-time quantum algorithm [CJS14] and the slow performance.

The recent revival of the GMW-FS design. Recently, there has been a revival of the study of the GMW-FS design, which is attributed to two research directions.

The first direction is the study of elliptic curve isogenies, following Couveignes and Stolbunov. As mentioned above, the issues here are mostly due to the computational aspects of group actions. To remedy this, the commutative group action CSIDH based on supersingular curves over prime fields was introduced in [CLM+18]. This led to the schemes SeaSign [FG19] and CSI-FiSh [BKV19], which greatly improve the situation by introducing both computational and protocol optimizations; see also the recent nice survey on this and more in [BFGP23].

The second direction may be viewed as a continuation of the polynomial isomorphism direction by Patarin [Pat96]. Three schemes submitted to the most recent NIST call for post-quantum digital signatures [oST22] fall into this category, namely LESS [BMPS20] based on linear code monomial equivalence, ATFE [TDJ+22] based on alternating trilinear form equivalence, and MEDS [CNP+23b] based on ma-

trix code equivalence[2]. Recent progress in complexity theory [GQ21b] shows that (1) linear code monomial equivalence reduces to matrix code equivalence in polynomial time [GQ21a, CDAG20], and (2) alternating trilinear form equivalence, isomorphism of quadratic polynomials with two secrets, cubic form equivalence, and matrix code equivalence are polynomial-time equivalent [GQ21b, GQT21] (see also [RST24] for some of these equivalences).

1.4 Overview of Contributions

In this thesis, the contributions are divided into three parts. In the first part, we design a practical digital signature scheme ALTEQ, its security only depends on the hardness of the alternating trilinear form equivalence (ATFE) problem. We also give new results on the complexity of ATFE. In the second part, we provide QROM security in a general framework based on group actions, and linkable ring signature under this framework. In part 3 we present state-of-the-art cryptanalysis for MEDS [CNP+23b] and ALTEQ.

Results about ATFE. In Chapter 2, we concern the complexity of testing equivalence of alternating trilinear forms. This problem is of interest in both mathematics and cryptography. We show that this problem is polynomial-time equivalent to testing equivalence of symmetric trilinear forms, by showing that they are both Tensor Isomorphism-complete, therefore is equivalent to test the isomorphism of cubic forms over most fields.

ALTEQ design and implementation. In Chapter 3, we turn our focus on constructing digital signature schemes based on ATFE. We propose a post-quantum signature scheme ALTEQ based on the ATFE problem. Our scheme is inspired by the GMW zero-knowledge interaction protocol [GMW91] for graph isomorphism. We implement the ALTEQ scheme with several optimizations, and submit it to NIST call for additional post-quantum signature standardization as the round 1 candidate.

[2]Matrix code equivalence is also known as 3-tensor isomorphism in [GQ21b].

QROM security and ring signature. In Chapter 4, we distill properties for group actions to be secure in the quantum random oracle model (QROM) based on the works [KLS18, LZ19, DFMS19]. We then apply these results to support the security of the ALTEQ scheme in the QROM model. In Chapter 5, we present the linkable ring signature construction of Beullens, Katsumata and Pintore [BKP20] with abstract group actions. We then apply the results to a concrete setting, namely the digital signature scheme ALTEQ. We implement the ring signature scheme above for ALTEQ and our preliminary experiments suggest that this scheme is competitive among existing post-quantum ring signatures.

Cryptanalysis on MEDS and ALTEQ. In Chapter 6, we outline the generic algorithms for MCE and ATFE at a high level, following Beullens [Beu23]. But in a departure from [Beu23] which relies on local invariants on graphs, we design new *global* invariants. We present new algorithms for MCE and ATFE in Chapter 7 and Chapter 8 respectively, which are further development of the algorithms for polynomial isomorphism and alternating trilinear form equivalence, in particular by Bouillaguet, Fouque, and Véber [BFV13], and Beullens [Beu23]. Key ingredients in these algorithms are new easy-to-compute distinguishing invariants under the respective group actions.

For MCE, we associate new isomorphism invariants to corank-1 points of matrix codes, which lead to a birthday-type algorithm. We present empirical justifications that these isomorphism invariants are easy-to-compute and distinguishing, and provide an implementation of this algorithm. This algorithm has some implications for the security of MEDS.

The invariant function for ATFE is similar, except it is associated with lower rank points. Modulo certain assumptions on turning the invariant function into canonical forms, our algorithm for ATFE improves on the runtime of the previously best known algorithm of Beullens [Beu23].

In the remaining sections of this chapter, we give detailed overviews of each of the results in this thesis.

1.5 Complexity of symmetric and alternating trilinear form equivalence

The polynomial isomorphism problem. Let $\mathbb{F}$ be a field, and let $X = \{x_1, \ldots, x_n\}$ be a set of variables. Let $\mathrm{GL}(n, \mathbb{F})$ be the general linear group consisting of $n \times n$ invertible matrices over $\mathbb{F}$. A natural group action of $A = (a_{i,j}) \in \mathrm{GL}(n, \mathbb{F})$ on the polynomial ring $\mathbb{F}[X]$ sends $f(x_1, \ldots, x_n)$ to $f \circ A := f(\sum_{j=1}^{n} a_{1,j}x_j, \ldots, \sum_{j=1}^{n} a_{n,j}x_j)$. The *polynomial isomorphism problem* (PI) asks, given $f, g \in \mathbb{F}[X]$, whether there exists $A \in \mathrm{GL}(n, \mathbb{F})$ such that $f = g \circ A$. In the literature, this problem was also called the polynomial equivalence problem [AS05].

An important subcase of PI is when the input polynomials are required to be homogeneous of degree d. In this case, this problem is referred to as the homogeneous polynomial isomorphism problem, denoted as d-HPI. Homogeneous degree-3 (resp. degree-2) polynomials are also known as cubic (resp. quadratic) forms.

From cubic forms to symmetric and alternating trilinear forms. In the context of polynomial isomorphism, cubic forms are of particular interest. In complexity theory, it was shown that d-HPI reduces to cubic form isomorphism over fields with dth roots of unity [AS05, AS06]. In multivariate cryptography, cubic form isomorphism also received special attention, since using higher degree forms results in less efficiency in the cryptographic protocols.

Just as quadratic forms are closely related with symmetric bilinear forms, cubic forms are closely related with symmetric trilinear forms. Let $\mathbb{F}$ be a field of characteristic not 2 or 3, and let $f = \sum_{1 \le i \le j \le k \le n} a_{i,j,k} x_i x_j x_k \in \mathbb{F}[x_1, \ldots, x_n]$ be a cubic form. For any $i, j, k \in [n]$, let $1 \le i' \le j' \le k' \le n$ be the result of sorting i, j, k in the

increasing order, and set $a_{i,j,k} = a_{i',j',k'}$. Then we can define a symmetric[3] trilinear form $\phi_f : \mathbb{F}^n \times \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by

$$\phi_f(u,v,w) = \sum_{i\in[n]} a_{i,i,i} u_i v_i w_i + \frac{1}{3} \cdot \sum_{\substack{i,j,k\in[n] \\ \text{two of } i,j,k \text{ are the same}}} a_{i,j,k} u_i v_j w_k + \frac{1}{6} \cdot \sum_{\substack{i,j,k\in[n] \\ i,j,k \text{ all different}}} a_{i,j,k} u_i v_j w_k.$$

It can be seen easily that for any $v = (v_1, \ldots, v_n)^{\mathrm{t}} \in \mathbb{F}^n$, $f(v_1, \ldots, v_n) = \phi_f(v, v, v)$.

In the theory of bilinear forms, symmetric and skew-symmetric bilinear forms are two important special subclasses. For example, they are critical in the classifications of classical groups [Wey97] and finite simple groups [Wil09b]. For trilinear forms, we also have skew-symmetric trilinear forms. In fact, to avoid some complications over fields of characteristics 2 or 3, we shall consider alternating trilinear forms which are closely related to skew-symmetric ones. For trilinear forms, the exceptional groups of type E_6 can be constructed as the stabilizer of certain symmetric trilinear forms, and those of type G_2 can be constructed as the stabilizer of certain alternating trilinear forms.

Definition 1.5.1 (Alternating trilinear form)**.** A trilinear form $\phi : \mathbb{F}^n \times \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ is *alternating*, if whenever two arguments of ϕ are equal, ϕ evaluates to zero.

Note that this implies skew-symmetry, namely for any $u_1, u_2, u_3 \in \mathbb{F}^n$ and any $\sigma \in \mathrm{S}_3$, $\phi(u_1, u_2, u_3) = \mathrm{sgn}(\sigma) \cdot \phi(u_{\sigma(1)}, u_{\sigma(2)}, u_{\sigma(3)})$. Over fields of characteristic zero or > 3, this is equivalent to skew-symmetry.

The trilinear form equivalence problem. Given a trilinear form $\phi : \mathbb{F}^n \times \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$, $A \in \mathrm{GL}(n, \mathbb{F})$ naturally acts on ϕ by sending it to $\phi \circ A := \phi(A^t(u), A^t(v), A^t(w))$. The *trilinear form equivalence problem* then asks, given two trilinear forms $\phi, \psi : \mathbb{F}^n \times \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$, whether there exists $A \in \mathrm{GL}(n, \mathbb{F})$, such that $\phi = \psi \circ A$. Over fields of characteristic not 2 or 3, two cubic forms f and g are isomorphic if and only if ϕ_f and ϕ_g are equivalent, so cubic form isomorphism is polynomial-time equivalent to symmetric trilinear form equivalence over such fields. Note that for clarity, we reserve

[3] That is, for any permutation $\sigma \in \mathrm{S}_3$, $\phi(u_1, u_2, u_3) = \phi(u_{\sigma(1)}, u_{\sigma(2)}, u_{\sigma(3)})$

the term "isomorphism" for polynomials (and cubic forms), and use "equivalence" for multilinear forms.

Motivations to study alternating trilinear form equivalence. Our main interest is to study the the complexity of alternating trilinear form equivalence, with the following motivations.

The first motivation comes from cryptography. To store a symmetric trilinear form on $\mathbb{F}_q^n$, $\binom{n+2}{3}$ field elements are required. To store an alternating trilinear form on $\mathbb{F}_q^n$, $\binom{n}{3}$ field elements are needed. The difference between $\binom{n+2}{3}$ and $\binom{n}{3}$ could be significant for practical purposes. For example, when $n = 9$, $\binom{n+2}{3} = \binom{11}{3} = 165$, while $\binom{n}{3} = \binom{9}{3} = 84$. This means that in the authentication protocol of Patarin [Pat96], using alternating trilinear forms instead of cubic forms for $n = 9$, saves almost one-half in the public key size, which is an important saving in practice.

The second motivation originates from comparing symmetric and alternating bilinear forms. It is well-known that, *in the bilinear case*, the structure of alternating forms is simpler than that of symmetric ones [Lan02]. Indeed, up to equivalence, an alternating bilinear form is completely determined by its rank over any field, while the classification of symmetric bilinear forms depends crucially on the underlying field. For example, recall that over $\mathbb{R}$, a symmetric form is determined by its "signature", so just the rank is not enough.

A third motivation is implied by the representation theory of the general linear groups; namely that alternating trilinear forms are the "last" natural case for $d = 3$. If we consider the action of $\mathrm{GL}(n, \mathbb{C})$ acting on d-tensors in $\mathbb{C}^n \otimes \mathbb{C}^n \otimes \cdots \otimes \mathbb{C}^n$ diagonally (that is, the same matrix acts on each tensor factor), it is a classical result [Wey97] that the invariant subspaces of $(\mathbb{C}^n)^{\otimes d}$ under this action are completely determined by the irreducible representations of $\mathrm{GL}(n, \mathbb{C})$. When $d = 3$, there are only three such representations, which correspond precisely to: symmetric trilinear forms, Lie algebras, and alternating trilinear forms. From the complexity point of view, it was previously shown that the isomorphism of symmetric trilinear forms [AS05, AS06] and

Lie algebras [GQ21b] are equivalent to algebra isomorphism. In Chapter 2, we show that the last case, the isomorphism of alternating trilinear forms, is also equivalent to the others.

Our Results. Given the above discussion on the comparison between symmetric and alternating bilinear forms, one may wonder whether alternating trilinear form equivalence was easier than symmetric trilinear form equivalence. Interestingly, we show that this is not the case; rather, they are polynomial-time equivalent.

Theorem 1.5.2. *The alternating trilinear form equivalence problem is polynomial-time equivalent to the symmetric trilinear form equivalence problem.*

Note here that the reduction from alternating to symmetric trilinear form equivalence requires us to go through the tensor isomorphism problem, which causes polynomial blow-ups in the dimensions of the underlying vector spaces. Therefore, though these two problems are polynomial-time equivalent, these problems may result in cryptosystems with different efficiencies for a given security level.

Relate Work. As mentioned in above, the degree-d homogeneous polynomial isomorphism problem (d-HPI) was shown to be almost equivalent to the algebra isomorphism problem (AI) in [AS05, AS06]. Here, almost refers to that for the reduction from d-HPI to AI in [AS05, AS06], the underlying fields are required to contain a dth root of unity. When $d = 3$, this means that the characteristic of the underlying field p satisfies that $p = 2 \mod 3$ or $p = 0$, which amounts to half of the primes. In [GQ21b], another reduction from 3-HPI to AI was presented, which works for fields of characteristics not 2 or 3. The reduction from AI to 3-HPI in [AS06] works over any field.

The tensor isomorphism complete class. In [FGS19, GQ21b], polynomial-time equivalences are proved between isomorphism testing of many more mathematical structures, including tensors, matrix spaces, polynomial maps, and so on. These problems arise from many areas: besides multivariate cryptography and computational complexity, they appear in quantum information, machine learning, and computa-

tional group theory. This motivates the authors of [GQ21b] to define the tensor isomorphism complete class TI, which we recall here.

Definition 1.5.3 (The d-Tensor Isomorphism problem, and the complexity class TI)**.** d-Tensor Isomorphism over a field $\mathbb{F}$ is the problem: given two d-way arrays $\mathrm{A} = (a_{i_1,\ldots,i_d})$ and $\mathrm{B} = (b_{i_1,\ldots,i_d})$, where $i_k \in [n_k]$ for $k \in [d]$, and $a_{i_1,\ldots,i_d}, b_{i_1,\ldots,i_d} \in \mathbb{F}$, decide whether there are $P_k \in \mathrm{GL}(n_k, \mathbb{F})$ for $k \in [d]$, such that for all $i_1, \ldots, i_d$,

$$a_{i_1,\ldots,i_d} = \sum_{j_1,\ldots,j_d} b_{j_1,\ldots,j_d} (P_1)_{i_1,j_1} (P_2)_{i_2,j_2} \cdots (P_d)_{i_d,j_d}. \tag{1.1}$$

For any field $\mathbb{F}$, $\mathrm{TI}_{\mathbb{F}}$ denotes the class of problems that are polynomial-time Turing (Cook) reducible to d-Tensor Isomorphism over $\mathbb{F}$, for some d. A problem is $\mathrm{TI}_{\mathbb{F}}$-complete, if it is in $\mathrm{TI}_{\mathbb{F}}$, and d-Tensor Isomorphism over $\mathbb{F}$ for any d reduces to this problem.

When a problem is naturally defined and is $\mathrm{TI}_{\mathbb{F}}$-complete over any $\mathbb{F}$, then we can simply write that it is TI-complete.

The authors of [GQ21b] further utilised this connection between tensors and groups to show search-to-decision, counting-to-decision, and nilpotency class results for p-group isomorphism [GQ21a]. It is worth mentioning that [GQ21b] shows the reduction from monomial code equivalence to tensor isomorphism. Interestingly, there is a concurrent and independent work [CDAG20] showing this reduction but using different techniques and terminology, specifically, tensor isomorphism is referred to matrix code equivalence in [CDAG20]. Moreover, matrix code equivalence is shown to be equivalent to the homogeneous version of the quadratic maps linear equivalence problem (QMLE) [RST24].

1.6 The ALTEQ signature scheme

In this thesis, we consider the *alternating trilinear form equivalence* (ATFE) problem. ATFE can be formulated as a Hidden Subgroup Problem (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 [GSVV04, MRS08, HMR+10]. This was termed by Moore, Russell, and Vazirani as "the strongest such insights we have about the limits of quantum algorithms" [MRV07]. As far as we know, this insight had not been used to *directly* support the security of any practical post-quantum cryptosystems. In this thesis, we will utilize this insight to investigate the practical use of ATFE in post-quantum cryptography.

Our Results. We propose and study a digital signature scheme based on the ATFE problem through the following steps.

(1) We propose a post-quantum signature scheme ALTEQ based on the ATFE problem. Our scheme is inspired by the GMW zero-knowledge interaction protocol [GMW91] for graph isomorphism. Our scheme is proven to be secure in the Random Oracle Model (ROM) based on the hardness of the ATFE problem.

(2) Based on the algorithmic study of ATFE in Chapter 8, we propose criteria for setting the parameters of these schemes to achieve a fixed security level in Section 3.5.1.

(3) We implement the ALTEQ with several optimizations for modular arithmetic, group actions, and seed expansion; see details in Section 3.5.2. We also provide AVX2 acceleration.

On interactions with other research lines. 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 [Pat96] 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 [FP06, BFFP11, BFV13]. Furthermore, some variants such as isomorphism of quadratic polynomials with one secret were shown to be easily solvable [BFFP11, Bou11, BFP15, IQ19].

For experts on isogeny-based cryptography, especially those who are familiar with SeaSign [FG19] and CSI-FiSh [BKV19], s/he would quickly recognize that our scheme has the same structure. The key difference lies in using a different action. The class group action as in CSIDH [CLM+18] 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.

1.7 On digital signatures based on group action: QROM security and ring signatures

Quantum Random Oracle Model (QROM). The random oracle model (ROM) was first proposed in 1993 by Bellare and Rogaway in [BR93] as a heuristic to provide security proofs in cryptography. Briefly speaking, in the ROM model, the hash function is modeled as by a random oracle. However, ROM is insufficient when considering quantum adversaries, which leads to the proposal of the *quantum* ROM (QROM) [BDF+11]. One main reason is that quantum adversaries can make queries at a superposition. For example, let $H : \mathcal{X} \to \mathcal{Y}$ be a hash function, a quantum adversary will make superposition queries to evaluate this function, that is, for input $\sum_x \beta_x |x\rangle$ return $\sum_x \beta_x |x\rangle|H(x)\rangle$. Security proof migration from ROM to QROM is not an easy task, due to several obstacles from some properties in the quantum setting, such as whether the query is a superposition, quantum no cloning, and quantum measurement causes collapse, etc. Indeed, there exist protocols that are secure in ROM but not in QROM [BDF+11, YZ21].

Linkable ring signatures. Ring signature, introduced by Rivest, Shamir and Tauman [RST01], is a special type of digital signature in which a signer can sign on behalf of a group chosen by him- or herself, while retaining anonymity within the group. In particular, ring signatures are formed without a complex setup procedure or the requirement for a group manager. They simply require users to be part of an existing public key infrastructure.

A linkable ring signature [LW05] is a variant of ring signatures in which any signatures produced by the same signer can be publicly linked. Linkable ring signatures are suitable in many different practical applications, such as privacy-preserving digital currency [SALY17] and e-voting [TW05].

Results for the GMW-FS design

In the following, we always let G denote a group, S a set, and $\alpha : G \times S \longrightarrow S$ a group action.

Security in the quantum random oracle model. The quantum random oracle model (QROM) was proposed by Boneh et al. [BDF+11] in 2011 and has received considerable attention since then. There are certain inherent difficulties in proving security in the QROM model, such as the adaptive programmability and rewinding [BDF+11]. Indeed, the QROM security of the Fiat–Shamir transformation was only recently shown after a series of works [Unr17, KLS18, LZ19, DFMS19].

In this thesis, we make progress on the QROM security of the GMW-FS design based on the works [Unr17, KLS18, LZ19, DFMS19]. Our results on this line can be informally summarised as follows. Recall that $\alpha : G \times S \longrightarrow S$ is a group action. In the GMW-FS design, the protocol starts with some (chosen or randomly sampled) set element $s \in S$. For $s \in S$, the stabilizer group $\mathrm{Stab}(s) := \{g \in G \mid \alpha(g, s) = s\}$.

(1) The GMW-FS scheme is secure in the QROM model, if $\mathrm{Stab}(s)$ is trivial, i.e. $|\mathrm{Stab}(s)| = 1$ and α satisfies the C-one-way-$\mathcal{O}(s)$ assumption (see Definition 4.1.2 and Remark 4.1.3).

(2) The GMW-FS scheme is secure in the QROM model, if the group action under ATFE satisfies the pseudorandom property as defined in [JQSY19, AFMP20] (see Definition 4.1.2), and the non-trivial automorphism hardness property (see Definition 4.2.6). In particular, in this setting the security proof is tight.

The GMW-FS-BKP ring signature design. Beullens, Katsumata and Pintore [BKP20] proposed an elegant way to construct efficient linkable ring signatures from group actions. Their focus was on commutative group actions, with instantiations in both isogeny and lattice settings. The advantage of their schemes is the scalability of signature sizes with the ring size, even compared to other logarithmic-size post-quantum ring signatures.

While [BKP20] focussed on commutative group actions, their ring signature construction is readily applicable to general group actions. In fact, for our group action framework, the scheme becomes a bit simpler because [BKP20] needs to work with rejection sampling due to certain stronger assumptions on the group actions. We call this ring signature design the GMW-FS-BKP design, and describe its construction in Section 5.2. The linkability property requires extra discussions as it calls for an interesting property of pairs of group actions.

Comparisons with some previous works. QROM securities and ring signature schemes have been shown for concrete schemes based on group actions. For example, the QROM security of CSI-FiSh (resp. MEDS, LESS) based on the perfect unique response was observed in [BKV19] (resp. [CNP+23b], [BMPS20]), and the tight QROM security based on a lossy version of CSI-FiSh was shown in [EKP20]. The ring signature scheme in [BKP20] has been shown for the group actions underlying CSI-FiSh [BKP20], LESS [BBN+22], and MEDS [CNP+23b].

Indeed, we view our results for the GMW-FS design as mostly conceptual. Our aim is to make these results convenient for future uses. That is, we distill properties of group actions (pairs) that are key to the QROM security (Definition 4.2.6) or for linkable ring signatures (Definition 5.3.2). We hope that these will not only

help with existing schemes, but also facilitate future schemes based on the GMW-FS design. Furthermore, to the best of our knowledge, the connection of the lossy approach for QROM security [KLS18] with the pseudorandom group action assumption [JQSY19, AFMP20] and the non-trivial automorphism hardness assumption (Definition 4.2.6) was not stated explicitly before. Such results should benefit the LESS and MEDS schemes, which only discussed their QROM securities based on perfect unique response (but not the lossy scheme).

Results for the ALTEQ scheme

After working with the general GMW-FS design, we focus on the ALTEQ scheme described in Chapter 3, which demonstrates concrete uses of the results we obtained for abstract group actions.

The QROM security of the ALTEQ scheme. Based on the results from the first part, there are two approaches to show its QROM security: the first is based on the automorphism group order statistics, and the second is based on the pseudorandom group action assumption. The sEUF-CMA security in QROM of ALTEQ scheme can be achieved by both two approaches.

For the first approach, we provide experimental results to support that, for those parameters proposed in Section 3.5.3, a random alternating trilinear form has the trivial automorphism group. This requires us to implement an algorithm for the automorphism group order computation.

For the second approach, the question of whether the group action under ATFE is pseudorandom or not is an open problem. In Section 3.3.2, some arguments were provided to support that it is. In particular, we do not need to modify the original ALTEQ scheme in Section 3.2.1 to attain the security in QROM, i.e., as opposed to the lossy CSI-FiSh scheme [EKP20]. We will discuss more about this in Section 1.7.

An implementation of the ring signature scheme for ALTEQ. We implement the ring signature protocol from [BKP20] for ALTEQ. Preliminary experimental results

suggest that it's more balanced than Calamari and Falafl in terms of signature size and signing time. We refer the reader to Section 5.4 and Table 1.1 for the details. Here we give a brief summary and comparison with some previous ring signature schemes.

Since we use the construction in [BKP20], the signature size of our schemes only depends on $\log R$, where R denotes the ring size. We see that our signature size can be estimated as $0.8 \log R + 19.7$KB, while the signature sizes of Calamari and Falafl in [BKP20] are estimated to be $\log R$+2.5KB and $0.5 \log R$+28.5KB respectively. For ring size $R = 8$, our signing time is 205ms which is twice Falafl's 90ms and much smaller than Calamari's 79s. Meanwhile, our ring signature size is 22.1KB, while Falafl and Calamari have the signature size of 30KB and 5.4KB respectively. RAPTOR [LAZ19], and DualRing-LB [YEL+21] have shorter signature sizes than ours when the ring size is small. However, their sizes are linearly dependent on the number of ring users; therefore, the size significantly increases when the number of participants rises. Regarding MRr-DSS [BESV22], while it performs well for low to medium users (<= 2^7), our protocol can outperform it in this range. For more comparisons with other ring signatures, please see Table 1.1. Finally, Fig 1.1 reports the signing time of our protocol; there, n, M and K are the parameters in the ring signature scheme for ALTEQ as defined in Section 5.4. Note that the signing time is measured on a 2.4 GHz Quad-Core Intel Core i5.

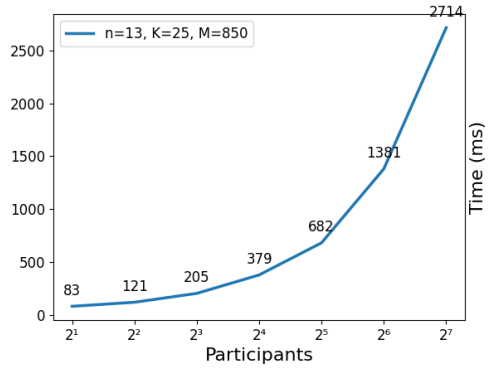

Figure 1.1: Signature generation time

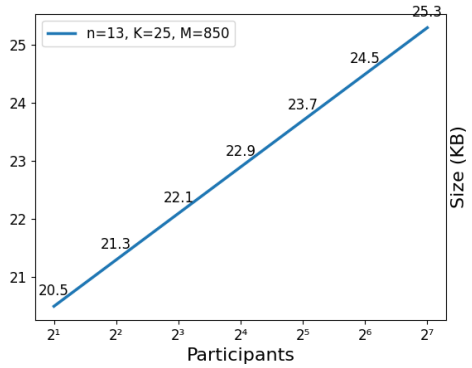

Figure 1.2: Signature size

	R								Hardness assumption	Secuirty level
	2^1	2^3	2^5	2^6	2^{10}	2^{12}	2^{15}	2^{21}		
MatRiCT [EZS+19]	18	19	/	31	/	59	/	148	MSIS, MLWE	128 bits
SMILE [LNS21]	/	/	16	/	18	/	19	/	MSIS, MLWE	128 bits
MatRiCT+ [ESZ22]	5.4	8.2	11	12.4	18	20.8	25	33.4	MSIS, MLWE	128 bits
RAPTOR [LAZ19]	2.5	10	/	81	/	5161	/	/	NTRU	100 bits
Calamari [BKP20]	3.5	5.4	/	8.2	/	14	/	23	CSIDH-512	*
Falafl [BKP20]	29	30	/	32	/	35	/	39	MSIS, MLWE	128 bits
DualRing-LB [YEL+21]	/	4.6	/	6	/	106.6	/	/	MSIS, MLWE	128 bits
MRr-DSS [BESV22]	/	27	/	36	/	422	/	/	MinRank	128 bits
LESS [BBN+22]	/	10.8	/	13.7	/	19.7	/	28.6	Code Equiv.	128 bits
Ours	20.5	22.1	23.7	24.5	27.7	29.3	31.7	36.5	ATFE	128 bits

Table 1.1: Comparison of the signature size (KB) between our schemes and others

Discussions on QROM security. The QROM security for the GMW-FS design was shown based on perfect unique responses and lossy schemes. There is one further approach that could avoid analyzing automorphism groups mathematically. In [LZ19, DFMS19], a property called *quantum unique response* in [DFMS19] or collapsing sigma protocol in [LZ19] is introduced, generalizing the *collapsingness* which is introduced by Unruh [Unr16] to the quantum setting. The definition of this property relies on a certain protocol and basically asks to distinguish between measuring or not measuring during the execution of the protocol. It is an interesting problem to study isomorphism problems from the point of this property, which would lead to another security proof under QROM.

Comparisons with results from isogeny-based cryptography. First, the group action underlying our lossy identification scheme is the same action as the basic AL-TEQ scheme as described in Section 3.2 , while the group action underlying the lossy CSI-FiSh [EKP20] is the diagonal action of the class group on two elliptic curves following [Sto12]. One reason is that for the pseudorandom group action assumption [JQSY19] (cf. Section 4.1.3) to be useful, it is necessary that the underlying group action is intransitive, but the class group action on the classes of elliptic curves are transitive, which is why two copies are needed there. This results in a doubling of

the public-key size in lossy CSI-FiSh compared to the original CSI-FiSh, as opposed to our case where the public key size remains the same.

Second, we compare the GMW-FS-BKP design applied to ATFE here with that of the class group action [BKP20]. The class group action leads to smaller signature sizes, but it suffers the problems of efficiently computing the group action and random sampling. The group action underlying ATFE allows for fast group action and random sampling, though the signature sizes are larger.

1.8 Algorithms for matrix code and alternating trilinear form equivalences

Background. Given two objects A and B of the same type, the *equivalence problem* asks if there exists a map π such that $\pi(A) = B$. The hardness of the equivalence problem depends on the objects and how the map is defined. There are objects in the equivalence problem that were recently proposed to support public-key cryptography for quantum-resistant purposes, such as linear or matrix codes [CNP+23b, BMPS20, BBN+22], alternating trilinear form (see Chapter 3), lattice [DvW22, DPPW22] etc.

Linear code equivalence. A classical equivalence problem is the *Code Equivalence* problem, which asks whether two given linear codes are isometric, that is, whether two linear codes are the same up to permuting, and possibly scalar multiplications on, the coordinates. One digital signature scheme submitted to the NIST call for additional signatures, LESS [BBB+23], is based on the assumed hardness of this problem.

Leon [Leo82] initiated the study of this problem and proposed an algorithm that computes a list of both codes with minimum Hamming weight and then matches them to recover the isometry. Recently, Beullens [Beu20] improved Leon's algorithm by using collision search. Another algorithm of significance is known as the Support Splitting Algorithm (SSA) by Sendrier [Sen00]. Its running time increases exponentially in the dimension of the hull (the intersection of a code and its dual), and it works effec-

tively for random linear codes under permutations. When scalar multiplications are also present, SSA works when $q \leqslant 4$ but not $q \geq 5$. If the hull is trivial and only permutations are used, then this problem can be reduced to graph isomorphism [BOST19].

Matrix code equivalence. In this work, we are interested in the equivalence problem of matrix codes, called the *Matrix Code Equivalence* (MCE) problem. A matrix code over $\mathbb{F}_q$ is a linear subspace of the space of $m \times n$ matrices over $\mathbb{F}_q$. Concerning the MCE problem, it was recently shown to be at least as hard as the Code Equivalence problem [CDAG20, GQ21b].

Alternating trilinear form equivalence. We are also interested in another problem namely *Alternating Trilinear Form Equivalence* (ATFE) problem. Here, the objects are alternating trilinear forms, namely a function $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ that is (1) linear in each argument, and (2) whenever two arguments are the same, ϕ evaluates to 0.

We now state the MCE problem and recall the ATFE problem, which would also indicate what equivalences mean for matrix codes and alternating trilinear forms.

Definition 1.8.1 (Matrix Code Equivalence (MCE))**.** Given two matrix codes C and $\mathcal{D}$ in $\mathrm{M}(m \times n, q)$, the problem asks whether there exist two invertible matrices $A \in \mathrm{GL}(m, q)$ and $B \in \mathrm{GL}(n, q)$ such that $\mathcal{D} = ACB := \{ACB \mid C \in C\}$.

Definition 1.8.2 (Alternating Trilinear Form Equivalence (ATFE))**.** Given two alternating trilinear forms $\phi, \psi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$, the problem asks whether there exists an invertible matrix $A \in \mathrm{GL}(n, q)$ such that for any $u, v, w \in \mathbb{F}_q^n$, $\phi(Au, Av, Aw) = \psi(u, v, w)$.

Relations between MCE and ATFE. MCE and ATFE are shown to be polynomial-time equivalent (see Chapter 2) and are Tensor Isomorphism (TI)-complete [GQ21b]. Based on the MCE and ATFE problems, two signature schemes have recently been proposed by Tang et al. [TDJ+22] and Chou et al. [CNP+23b]. Subsequently, various applications have arisen, including ring signatures [BCD+22, DG22, CNP+23b] and threshold signatures [BBMP23]. These works lead to submissions to NIST's current stan-

dardization for post-quantum signatures: MEDS [CNP+23b] and ALTEQ [BDN+23]. Hence, it is of significance to investigate the hardness of these two problems, as it will provide insights into the selection of secure parameter sets.

Previous work. We will briefly review some of the state-of-the-art algorithms for MCE and ATFE. Algorithms for MCE and ATFE have been surveyed in [CNP+23b] and [TDJ+22], respectively. Beullens recently contributed beautiful new algorithms for ATFE in [Beu23]. Here we explain two algorithms, one for MCE and one for ATFE, that are most relevant to us.

Leon-like algorithm for MCE. Leon's algorithm [Leo82] is well-known for solving code equivalence problem in the Hamming metric. The key observation is that the equivalence preserves the hamming weight of the codewords. Consequently, identifying the set of codewords with minimum hamming weight within two codes can aid in revealing the equivalence or isometry between the codes. Recently, Beullens [Beu20] improved upon this algorithm by constructing the set of codewords with particular weight and the same multiset of entries as lists [4]. Subsequently, a collision search is conducted between the two lists to recover equivalence or isometry easily. It is natural to adapt Leon's algorithm to MCE [CNP+23b]. That is, one can first build two lists of low-rank matrices in C_1 and C_2, and then do a collision search to find a matched pair of corresponding matrix codes and so recover the equivalence.

Beullens' algorithm for ATFE. Beullens [Beu23] currently proposed a graph-theoretic algorithm to solve the ATFE problem. An alternating trilinear form ϕ can be viewed as a graph G_ϕ, where $\mathbf{v} \in \mathbb{F}_q^n$ is a vertex and $(\mathbf{u}, \mathbf{v})$ be an edge if and only if $\phi_{\mathbf{u},\mathbf{v}} = 0$. Also, a binlinear form $\phi_{\mathbf{u}}$ can be viewed as a matrix $M_{\phi,\mathbf{u}}$, then the rank of $\mathbf{u}$ is the rank of $M_{\phi,\mathbf{u}}$. The key observation is that the equivalence preserves the rank of the vertices in G_ϕ. Therefore, the algorithm first builds two lists of low-rank points in ϕ and ψ respectively and then finds a collision to recover the equivalence.

[4] In the monomial setting, Beullens considered building a set of 2-dimension subcodes with small support. This is because monomial transformation do not preserve anything beyond the hamming weight of a vector.

parameter set	n	q	**Algebraic**	**Leon-like**	**Ours**
MEDS-I	14	4093	148.1	170.68	102.59
MEDS-III	22	4093	218.41	246.95	152.55
MEDS-V	30	2039	298.82	297.77	186.57

Table 1.2: Algorithms for solving the MCE problem. The data for algebraic and Leon-like algorithms are from the MEDS specification [CNP+23b].

Gröbner basis approach. The MCE and ATFE problem can be solved algebraically by transforming them into a system of polynomial equations and then solving this system via Gröbner basis [TDJ+22, CNP+23b]. The Gröbner basis method, exhibits insensitivity to the parameter q within the system, with its efficiency contingent solely upon the values m, n and l (or n for the ATFE). Also, this approach demonstrates high efficiency when applied to problems characterized by low dimensions.

Our results. We propose two heuristic algorithms for MCE and ATFE problems, respectively. We summarize our contributions below.

Algorithm for MCE. We present a new algorithm for MCE. Our algorithm introduces a novel invariant for matrix codes, which we call the "corank-1 associated invariant". This innovation allows us to find a collision using the birthday paradox, and it avoids the use of Gröbner basis computations. This improvement leads to an algorithm with a complexity of $O(q^{(n-2)/2} \cdot (q \cdot n^3 + n^4) \cdot (\log(q))^2)$ as described in Section 7.3. We provide an implementation of this algorithm, and demonstrate its practical effectiveness for small n and q (such as $n = 9$ and $q = 31$) in Section 7.5.

Regarding the MEDS scheme, its security is based on the hardness of the MCE problem. Although our algorithm does not yet achieve a practical break of the parameter sets proposed by MEDS, it serves to underscore that these parameters have not yet attained the target security level; see Table 1.2.

Importantly, we note that this could be fixed easily by enlarging q. This fix should not affect the running times, and only increase the signature sizes *at most*[5] linearly in

[5]It is at most because of the use of the seed tree techniques; see [CNP+23a] for more details.

$\log(q)$. Therefore the consequence of our algorithm on MEDS should be considered as mild.

Algorithm for ATFE. We present an algorithm for the ATFE problem by introducing a novel isomorphism invariant. For an alternating trilinear form ϕ and a low-rank point v, the equivalence preserves the kernel space K of v. Based on this observation, we define the isomorphism invariant as a new alternating trilinear form $\hat{\phi}$ under the action of $\mathrm{GL}(K)\times\mathrm{GL}(n,q)$. This isomorphism invariant leads to the birthday algorithm to find a collision, resulting in an algorithm with complexity with the dominating factor being $O(q^{k/2})$, as opposed to the algorithms in [Beu23] with the dominating factors being $O(q^k)$ or $O(q^{n/2})$. This algorithm was used to determine the parameters of ALTEQ [BDN+23].

Our algorithms as a further development of [BFV13,Beu23]. Our algorithms for MCE and ATFE follow the previous works on polynomial isomorphism and alternating trilinear form equivalence. In particular, our algorithms are a further development of the works of Bouillaguet, Fouque, and Véber [BFV13], and Beullens [Beu23].

In [BFV13], algorithms for testing isomorphism of systems of quadratic forms were presented. Both algorithms rely on certain graphs associated with quadratic form systems. The first algorithm in [BFV13] samples a list of low-rank points for each of the two input polynomial systems, and finds a collision that can be used in conjunction of the hybrid Gröbner basis method [FP06] to recover the secret transformation. The second algorithm in [BFV13] works for $q = 2$; it is based on birthday paradox with an isomorphism invariant obtained by examining the radius-k neighborhood of the points in the graph.

In [Beu23], algorithms for ATFE were presented. Two of the algorithms that are most relevant to us are as follows. (We refer the reader to [Beu23] for a beautiful algorithm for $n = 9$.) The first algorithm follows the sampling and collision approach, with the main innovation being that for the sampling step, where Beullens uses a random walk on the graph associated with an alternating trilinear form. The second

algorithm is based on the birthday paradox with isomorphism invariants. As q is large for the use of ATFE in [TDJ+22], Beullens used radius-1 or -2 neighborhoods and observed that such neighborhood information is distinguishing.

Our algorithms for MCE and ATFE are based on the birthday paradox with isomorphism invariants (see Section 6.2). As seen from the above, previous works use isomorphism invariants that are *local* (small radius neighborhood) on graphs associated with polynomial systems or trilinear forms. Our main technical contribution is to discover new isomorphism invariants that can be viewed as transforming the information from graphs to *global* constraints.

For example, the isomorphism invariants for MCE are obtained by associating some graphs with matrix codes. We also perform a walk on the graph (starting from a corank-1 point), but we then use the path information to transform the matrix code as a whole to obtain an isomorphism invariant. Similarly, for ATFE, the isomorphism invariants are obtained by first taking the kernel of a low-rank point. We then apply this kernel to the alternating trilinear form to obtain another (smaller) trilinear form, and use this trilinear form as an isomorphism invariant.

1.9 Publications and works contained in this thesis

The results in this thesis are based on the following works [6]:

(1) *Average-case algorithms for testing isomorphism of polynomials, algebras, and multilinear forms* [GQT21], with Joshua A Grochow and Youming Qiao (STACS 2021).

(2) *Practical post-quantum signature schemes from isomorphism problems of trilinear forms* [TDJ+22], with Dung Hoang Duong, Antoine Joux, Thomas Plantard, Youming Qiao and Willy Susilo (Eurocrypt 2022).

[6] Note for the author order: I am the first author of (2), and the authors of the remaining works are all sorted in alphabetical order.

(3) *ALTEQ: Digital Signatures from Alternating Trilinear Form Equivalence* [BDN+23], with Markus Bläser, Dung Hoang Duong, Anand Kumar Narayanan, Thomas Plantard, Youming Qiao and Arnaud Sipasseuth (Round 1 candidate of NIST Additional Call for Post-quantum Digital Signature Schemes).

(4) *On digital signatures based on isomorphism problems: QROM security and ring signatures* [BCD+22], with Markus Bläser, Zhili Chen, Dung Hoang Duong, Antoine Joux, Ngoc Tuong Nguyen, Thomas Plantard, Youming Qiao and Willy Susilo (PQCrypto 2024).

(5) *Algorithms for matrix code and trilinear form equivalences via new isomorphism invariants*, with Anand Kumar Narayanan and Youming Qiao (Eurocrypt 2024).

Other publications not included in this thesis:

(1) *On the complexity of isomorphism problems for tensors, groups, and polynomials III: actions by classical groups.* [CGQ+24], with Zhili Chen, Joshua A. Grochow, Youming Qiao and Chuanqi Zhang (ITCS 2024).

Part I

Digital signatures from alternating trilinear form equivalence

Chapter 2

Complexity of alternating trilinear form equivalence

2.1 Technical overview

We recall the Theorem 1.5.2 there, that is, the alternating trilinear form equivalence problem is polynomial-time equivalent to the symmetric trilinear form equivalence problem.

Techniques for proving Theorem 1.5.2. By [FGS19], the trilinear form equivalence problem is in Tensor Isomorphism (TI), and so are the special cases symmetric and alternating trilinear form equivalence. The proof of Theorem 1.5.2 goes by showing that both symmetric and alternating trilinear form equivalence are TI-hard.

Technically, the basic proof strategy is to adapt a gadget construction, which originates from [FGS19] and then is further used in [GQ21b]. To use that gadget in the trilinear form setting does require several non-trivial ideas. First, we identify the right TI-complete problem to start with, namely the alternating (resp. symmetric) matrix space isometry problem. Second, we need to arrange a 3-way array A, representing a linear basis of an alternating (resp. symmetric) matrix spaces, into one representing an alternating trilinear form. This requires 3 copies of A, assembled in an appropri-

ate manner. Third, we need to add the gadget in three directions (instead of just two as in previous results). All these features were not present in [FGS19, GQ21b]. The correctness proof also requires certain tricky twists compared with those in [FGS19] and [GQ21b].

2.2 Chapter preliminaries

Notations. Let $\mathbb{F}$ be a field. Vectors in $\mathbb{F}^n$ are column vectors. Let e_i denote the ith standard basis vector of $\mathbb{F}^n$. Let $\mathrm{M}(\ell \times n, \mathbb{F})$ be the linear space of $\ell \times n$ matrices over $\mathbb{F}$, and set $\mathrm{M}(n, \mathbb{F}) := \mathrm{M}(n \times n, \mathbb{F})$. Let I_n denote the identity matrix of size n. For $A \in \mathrm{M}(n, \mathbb{F})$, A is *symmetric* if $A^{\mathrm{t}} = A$, and *alternating* if for every $v \in \mathbb{F}^n$, $v^{\mathrm{t}}Av = 0$. When the characteristic of $\mathbb{F}$ is not 2, A is alternating if and only if A is skew-symmetric. Let $\mathrm{S}(n, \mathbb{F})$ be the linear space of $n \times n$ symmetric matrices over $\mathbb{F}$, and let $\Lambda(n, \mathbb{F})$ be the linear space of alternating matrices over $\mathbb{F}$. When $\mathbb{F} = \mathbb{F}_q$, we may write $\mathrm{M}(n, \mathbb{F}_q)$ as $\mathrm{M}(n, q)$. We use $\langle \cdot \rangle$ to denote the linear span.

3-way arrays. A 3-way array over a field $\mathbb{F}$ is an array with three indices whose elements are from $\mathbb{F}$. We use $\mathrm{M}(n_1 \times n_2 \times n_3, \mathbb{F})$ to denote the linear space of 3-way arrays of side lengths $n_1 \times n_2 \times n_3$ over $\mathbb{F}$.

Let $\mathrm{A} \in \mathrm{M}(\ell \times n \times m, \mathbb{F})$. For $k \in [m]$, the kth *frontal* slice of A is $(a_{i,j,k})_{i \in [\ell], j \in [n]} \in \mathrm{M}(\ell \times n, \mathbb{F})$. For $j \in [n]$, the jth *vertical* slice of A is $(a_{i,j,k})_{i \in [\ell], k \in [m]} \in \mathrm{M}(\ell \times m, \mathbb{F})$. For $i \in [\ell]$, the ith *horizontal* slice of A is $(a_{i,j,k})_{j \in [n], k \in [m]} \in \mathrm{M}(n \times m, \mathbb{F})$. We shall often think of A as a matrix tuple in $\mathrm{M}(\ell \times n, \mathbb{F})^m$ consisting of its frontal slices.

A natural action of $(P, Q, R) \in \mathrm{GL}(\ell, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F}) \times \mathrm{GL}(m, \mathbb{F})$ sends a 3-way array $\mathrm{A} \in \mathrm{M}(\ell \times n \times m, \mathbb{F})$ to $P^{\mathrm{t}}\mathrm{A}^R Q$, defined as follows. First represent A as an m-tuple of $\ell \times n$ matrices $\mathbf{A} = (A_1, \ldots, A_m) \in \mathrm{M}(\ell \times n, \mathbb{F})^m$. Then P and Q send $\mathbf{A}$ to $P^{\mathrm{t}}\mathbf{A}Q = (P^{\mathrm{t}}A_1Q, \ldots, P^{\mathrm{t}}A_mQ)$, and $R = (r_{i,j})$ sends $\mathbf{A}$ to $(A'_1, \ldots, A'_m)$ where $A'_i = \sum_{j \in [m]} r_{i,j} A_j$. Clearly, the actions of P, Q, and R commute. The resulting m-tuple of $\ell \times n$ matrices obtained by applying P, Q, and R to $\mathbf{A}$ is then $P^{\mathrm{t}}\mathbf{A}^R Q$. Note

that up to possibly relabelling indices, the entries of $P^{\mathrm{t}}\mathbf{A}^R Q$ are explicitly defined as in Equation 1.1.

Matrix tuples. Let $\mathbf{A} = (A_1, \ldots, A_m), \mathbf{B} = (B_1, \ldots, B_m) \in \mathrm{M}(n, \mathbb{F})^m$. Given $T \in \mathrm{GL}(n, \mathbb{F})$, let $T^{\mathrm{t}}\mathbf{A}T = (T^{\mathrm{t}}A_1T, \ldots, T^{\mathrm{t}}A_mT)$. We say that $\mathbf{A}$ and $\mathbf{B}$ are *isometric*, if there exists $T \in \mathrm{GL}(n, \mathbb{F})$ such that $T^{\mathrm{t}}\mathbf{A}T = \mathbf{B}$. Let $\mathrm{Iso}(\mathbf{A}, \mathbf{B}) = \{T \in \mathrm{GL}(n, \mathbb{F}) : \mathbf{A} = T^{\mathrm{t}}\mathbf{B}T\}$, and set $\mathrm{Aut}(\mathbf{A}) := \mathrm{Iso}(\mathbf{A}, \mathbf{A})$. Clearly, $\mathrm{Aut}(\mathbf{A})$ is a subgroup of $\mathrm{GL}(n, q)$, and $\mathrm{Iso}(\mathbf{A}, \mathbf{B})$ is either empty or a coset of $\mathrm{Aut}(\mathbf{A})$.

2.3 The ATFE problem is TI-hard.

As mentioned in Section 2.1, the proof of Theorem 1.5.2 follows by showing that symmetric and alternating trilinear form equivalence are TI-hard (recall Definition 1.5.3). In the following we focus on the alternating case. The symmetric case can be tackled in a straightforward way, by starting from the TI-complete problem, symmetric matrix tuple pseudo-isometry, from [GQ21b, Theorem B], and modifying the alternating gadget to a symmetric one.

Proposition 2.3.1. *The alternating trilinear form equivalence problem is* TI*-hard.*

Proof. **The starting TI-complete problem.** We use the following TI-complete problem from [GQ21b]. Let $\mathbf{A} = (A_1, \ldots, A_m), \mathbf{B} = (B_1, \ldots, B_m) \in \Lambda(n, \mathbb{F})^m$ be two tuples of alternating matrices. We say that $\mathbf{A}$ and $\mathbf{B}$ are pseudo-isometric, if there exist $C \in \mathrm{GL}(n, \mathbb{F})$ and $D = (d_{i,j}) \in \mathrm{GL}(m, \mathbb{F})$, such that for any $i \in [m]$, $C^{\mathrm{t}}(\sum_{j\in[m]} d_{i,j}A_j)C = B_i$. By [GQ21b, Theorem B], the alternating matrix tuple pseudo-isometry problem is TI-complete. Without loss of generality, we assume that $\dim(\langle A_i \rangle) = \dim(\langle B_i \rangle)$, as if not, then they cannot be pseudo-isometric, and this dimension condition is easily checked.

An alternating trilinear form $\phi : \mathbb{F}^n \times \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ naturally corresponds to a 3-way array $\mathrm{A} = (a_{i,j,k}) \in \mathrm{M}(n \times n \times n, \mathbb{F})$, where $a_{i,j,k} = \phi(e_i, e_j, e_k)$. Then A is also alternating, i.e. $a_{i,j,k} = 0$ if $i = j$ or $i = k$ or $j = k$, and $a_{i,j,k} = \mathrm{sgn}(\sigma)a_{\sigma(i),\sigma(j),\sigma(k)}$

for any $\sigma \in S_3$. So in the following, we present a construction of an alternating 3-way array from an alternating matrix tuple, in such a way that two alternating matrix tuples are pseudo-isometric if and only if the corresponding alternating trilinear forms are equivalent.

Constructing alternating 3-way arrays from alternating matrix tuples. Given $\mathbf{A} = (A_1, \ldots, A_m) \in \Lambda(n, \mathbb{F})^m$, we first build the $n \times n \times m$ tensor A which has $A_1, \ldots, A_m$ as its frontal slices. Then we will use essentially the following construction twice in succession. We will give two viewpoints on this construction: one algebraic, in terms of trilinear forms, and another "matricial", in terms of 3-way arrays. Different readers may prefer one viewpoint over the other; our opinion is that the algebraic view makes it easier to verify the alternating property while the matricial view makes it easier to verify the reduction. The construction is, in some sense, the 3-tensor analogue of taking an ordinary matrix A and building the alternating matrix $\begin{bmatrix} 0 & A \\ -A^{\mathrm{t}} & 0 \end{bmatrix}$.

Notation: Just as the transpose acts on matrices by $(A^{\mathrm{t}})_{i,j} = A_{j,i}$, for a 3-tensor A, we have six possible "transposes" corresponding to the six permutations of the three coordinates. Given $\sigma \in S_3$, we write A^{σ} for the 3-tensor defined by $(\mathrm{A}^{\sigma})_{i_1,i_2,i_3} = \mathrm{A}_{i_{\sigma(1)},i_{\sigma(2)},i_{\sigma(3)}}$.

Given a 3-way array $\mathrm{A} \in \mathrm{M}(n \times m \times d, \mathbb{F})$, we will make use of $\mathrm{A}^{(23)}$ and $\mathrm{A}^{(13)}$:

- $\mathrm{A}^{(23)}$ is $n \times d \times m$ and has $\mathrm{A}^{(23)}_{i,j,k} = \mathrm{A}_{i,k,j}$. Equivalently, the k-th frontal slice of $\mathrm{A}^{(23)}$ is the k-th vertical slice of A.
- $\mathrm{A}^{(13)}$ is $d \times m \times n$ and has $\mathrm{A}^{(13)}_{i,j,k} = \mathrm{A}_{k,j,i}$. Equivalently, the k-th frontal slice of $\mathrm{A}^{(13)}$ is the transpose of the k-th horizontal slice of A.

Example 2.3.2 (Running example)**.** Let us examine a simple example as follows. Let $\mathbf{A} = (A) \in \Lambda(2, \mathbb{F})^1$, where $A = \begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix}$. Then $\mathrm{A} = (A)$; $\mathrm{A}^{(23)} = (A'_1, A'_2) \in \mathrm{M}(2 \times$

$1 \times 2, \mathbb{F})$, where $A'_1 = \begin{bmatrix} 0 \\ -a \end{bmatrix}$, and $A'_2 = \begin{bmatrix} a \\ 0 \end{bmatrix}$; $\mathrm{A}^{(13)} = (A''_1, A''_2) \in \mathrm{M}(1 \times 2 \times 2, \mathbb{F})$, where $A''_1 = \begin{bmatrix} 0 & a \end{bmatrix}$, and $A''_2 = \begin{bmatrix} -a & 0 \end{bmatrix}$.

From the above A, $\mathrm{A}^{(23)}$, and $\mathrm{A}^{(13)}$, we construct $\tilde{\mathrm{A}} \in \mathrm{M}((n+m)\times(n+m)\times(n+m), \mathbb{F})$ as follows. We divide $\tilde{\mathrm{A}}$ into the following eight blocks. That is, set $\tilde{\mathrm{A}} = (\tilde{\mathrm{A}}_1, \tilde{\mathrm{A}}_2)$ (two block frontal slices) where $\tilde{\mathrm{A}}_1 = \begin{bmatrix} 0_{n\times n\times n} & \mathrm{A}^{(23)} \\ \mathrm{A}^{(13)} & 0 \end{bmatrix}$, and $\tilde{\mathrm{A}}_2 = \begin{bmatrix} -\mathrm{A} & 0 \\ 0 & 0_{m\times m\times m} \end{bmatrix}$, where $0_{n\times n\times n}$ indicates the $n \times n \times n$ zero tensor, and analogously for $0_{m\times m\times m}$ (the remaining sizes can be determined from these and the fact that A is $n \times n \times m$).

The corresponding construction on trilinear forms is as follows. The original trilinear form is $A(x, y, z) = \sum_{i,j\in[n],k\in[m]} a_{i,j,k} x_i y_j z_k$, where $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n)$, and $z = (z_1, \ldots, z_m)$, and we have $A(x, y, z) = -A(y, x, z)$. The new trilinear form will be $\tilde{A}(x', y', z')$, where

$$\begin{aligned} x' = (x^{(1)}, x^{(2)}) &= (x_1^{(1)}, \ldots, x_n^{(1)}, x_1^{(2)}, \ldots, x_m^{(2)}) \\ y' = (y^{(1)}, y^{(2)}) &= (y_1^{(1)}, \ldots, y_n^{(1)}, y_1^{(2)}, \ldots, y_m^{(2)}) \\ z' = (z^{(1)}, z^{(2)}) &= (z_1^{(1)}, \ldots, z_n^{(1)}, z_1^{(2)}, \ldots, z_m^{(2)}). \end{aligned}$$

This new form will satisfy $\tilde{A}(x', y', z') = \sum_{i,j,k\in[n+m]} \tilde{a}_{i,j,k} x'_i y'_j z'_k$. Let us unravel what this looks like from the above description of $\tilde{\mathrm{A}}$. We have

$$\begin{aligned} \tilde{A}(x', y', z') &= \sum_{i\in[n],j\in[m],k\in[n]} (\tilde{\mathrm{A}}_1)_{i,n+j,k} x'_i y'_{n+j} z'_k + \sum_{i\in[m],j,k\in[n]} (\tilde{\mathrm{A}}_1)_{n+i,j,k} x'_{n+i} y'_j z'_k \\ &\quad + \sum_{i,j\in[n],k\in[m]} (\tilde{\mathrm{A}}_2)_{i,j,k} x'_i y'_j z'_{n+k} \\ &= \sum_{i\in[n],j\in[m],k\in[n]} \mathrm{A}^{(23)}_{i,j,k} x'_i y'_{n+j} z'_k + \sum_{i\in[m],j,k\in[n]} \mathrm{A}^{(13)}_{i,j,k} x'_{n+i} y'_j z'_k - \sum_{i,j\in[n],k\in[m]} \mathrm{A}_{i,j,k} x'_i y'_j z'_{n+k} \\ &= \sum_{i\in[n],j\in[m],k\in[n]} \mathrm{A}_{i,k,j} x'_i y'_{n+j} z'_k + \sum_{i\in[m],j,k\in[n]} \mathrm{A}_{k,j,i} x'_{n+i} y'_j z'_k - \sum_{i,j\in[n],k\in[m]} \mathrm{A}_{i,j,k} x'_i y'_j z'_{n+k} \\ &= A(x^{(1)}, z^{(1)}, y^{(2)}) + A(z^{(1)}, y^{(1)}, x^{(2)}) - A(x^{(1)}, y^{(1)}, z^{(2)}) \end{aligned}$$

From this formula, and the fact that $A(x, y, z) = -A(y, x, z)$, we can now more easily verify that $\tilde{A}$ is alternating in all three arguments. Since the permutations (13) and

(23) generate S_3, it suffices to verify it for these two. We have

$$\begin{aligned}
\tilde{A}^{(13)}(x', y', z') &= \tilde{A}(z', y', x') \\
&= A(z^{(1)}, x^{(1)}, y^{(2)}) + A(x^{(1)}, y^{(1)}, z^{(2)}) - A(z^{(1)}, y^{(1)}, x^{(2)}) \\
&= -A(x^{(1)}, z^{(1)}, y^{(2)}) + A(x^{(1)}, y^{(1)}, z^{(2)}) - A(z^{(1)}, y^{(1)}, x^{(2)}) \\
&= -\tilde{A}(x', y', z').
\end{aligned}$$

Similarly, we have:

$$\begin{aligned}
\tilde{A}^{(23)}(x', y', z') &= \tilde{A}(x', z', y') \\
&= A(x^{(1)}, y^{(1)}, z^{(2)}) + A(y^{(1)}, z^{(1)}, x^{(2)}) - A(x^{(1)}, z^{(1)}, y^{(2)}) \\
&= A(x^{(1)}, y^{(1)}, z^{(2)}) - A(z^{(1)}, y^{(1)}, x^{(2)}) - A(x^{(1)}, z^{(1)}, y^{(2)}) \\
&= -\tilde{A}(x', y', z'),
\end{aligned}$$

as claimed.

Example 2.3.3 (Running example, continued from Example 2.3.2)**.** We can write out $\tilde{\mathrm{A}}$ in this case explicitly. The first block frontal slice $\tilde{\mathrm{A}}_1$ is $3 \times 3 \times 2$, consisting of the two frontal slices

$$\left(\begin{array}{cc:c} 0 & 0 & 0 \\ 0 & 0 & -a \\ \hdashline 0 & a & 0 \end{array}\right) \text{ and } \left(\begin{array}{cc:c} 0 & 0 & a \\ 0 & 0 & 0 \\ \hdashline -a & 0 & 0 \end{array}\right)$$

while the second block frontal slice $\tilde{\mathrm{A}}_2$ is the $3 \times 3 \times 1$ matrix

$$\left(\begin{array}{cc:c} 0 & -a & 0 \\ a & 0 & 0 \\ \hdashline 0 & 0 & 0 \end{array}\right)$$

It can be verified easily that $\tilde{\mathrm{A}} = (a_{i,j,k})$ is alternating: the nonzero entries are $a_{2,3,1} = -a$, $a_{3,2,1} = a$, $a_{1,3,2} = a$, $a_{3,1,2} = -a$, $a_{1,2,3} = -a$, and $a_{2,1,3} = a$, which are consistent with the signs of the permutations.

The gadget construction. We now describe the gadget construction. The gadget can be described as a block 3-way array as follows. Construct a 3-way array G of size $(n+1)^2 \times (n+1)^2 \times (n+m)$ over $\mathbb{F}$ as follows. For $i \in [n]$, the ith frontal slice of G is

$$\begin{bmatrix} 0 & 0 & \dots & 0 & I_{n+1} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ -I_{n+1} & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix},$$

where 0 here denotes the $(n+1) \times (n+1)$ all-zero matrix, I_{n+1} is at the $(1, i+1)$th block position, and $-I_{n+1}$ is at the $(i+1, 1)$th block position. For $n+1 \le i \le n+m$, the ith frontal slice of G is the all-zero matrix. We also need the following 3-way arrays derived from G. We will use $\mathrm{G}^{(13)}$ and $\mathrm{G}^{(23)}$. Note that $\mathrm{G}^{(13)}$ is of size $(n+m) \times (n+1)^2 \times (n+1)^2$, and its ith horizontal slice is the ith frontal slice of G. Similarly, $\mathrm{G}^{(23)}$ is of size $(n+1)^2 \times (n+m) \times (n+1)^2$, and its jth vertical slice is the jth frontal slice of G.

Finally, construct a 3-tensor $\hat{\mathrm{A}}$ as follows. It consists of the two block frontal slices

$$\begin{bmatrix} \tilde{\mathrm{A}} & 0 \\ 0 & -\mathrm{G} \end{bmatrix} \text{ and } \begin{bmatrix} 0 & \mathrm{G}^{(13)} \\ \mathrm{G}^{(23)} & 0 \end{bmatrix}.$$

To see how this all fits together, let G_1 be the $(n+1)^2 \times (n+1)^2 \times n$ tensor consisting of the first n frontal slices of G (these are the only nonzero frontal slices of G). Then we may view $\hat{\mathrm{A}}$ as having three block frontal slices, namely:

$$\begin{bmatrix} 0_{n\times n\times n} & \mathrm{A}^{(23)} & 0 \\ \mathrm{A}^{(13)} & 0_{m\times m\times n} & 0 \\ 0 & 0 & -\mathrm{G}_1 \end{bmatrix}, \begin{bmatrix} -\mathrm{A} & 0 & 0 \\ 0 & 0_{m\times m\times m} & 0 \\ 0 & 0 & 0_{(n+1)^2\times(n+1)^2\times m} \end{bmatrix},$$

and

$$\begin{bmatrix} 0_{n\times n\times (n+1)^2} & 0 & G_1^{(13)} \\ 0 & 0_{m\times m\times (n+1)^2} & 0 \\ G_1^{(23)} & 0 & 0 \end{bmatrix}.$$

We claim that $\hat{A}$ is alternating. To verify this is straightforward but somewhat tedious. So we use the following example from which a complete proof can be extracted easily.

Example 2.3.4 (Running example, continued from Example 2.3.3)**.** Let A be the $2\times2\times1$ tensor with alternating frontal slice $A = \begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix}$. In particular, $n = 2, m = 1$, so G will have size $(n+1)^2\times(n+1)^2\times(n+m) = 9\times9\times3$, and A will have size $n+m+(n+1)^2 = 12$ in all three directions. We will write out the first $n + m = 3$ frontal slices explicitly, as those are the only ones involving A, and leave the last 9 (involving only transposes of G_1) unwritten.

$$\left(\begin{array}{ccc:c:ccc} 0 & 0 & 0 & & & & \\ 0 & 0 & -a & & & & \\ \hdashline 0 & a & 0 & & & & \\ \hdashline & & & 0_3 & I_3 & 0 \\ & & & -I_3 & 0_3 & 0 \\ & & & 0 & 0 & 0_3 \end{array}\right), \left(\begin{array}{ccc:c:ccc} 0 & 0 & a & & & & \\ 0 & 0 & 0 & & & & \\ \hdashline -a & 0 & 0 & & & & \\ \hdashline & & & 0_3 & 0 & I_3 \\ & & & 0 & 0_3 & 0 \\ & & & -I_3 & 0 & 0_3 \end{array}\right),$$

and

$$\left(\begin{array}{cc:c:ccc} 0 & a & 0 & & & \\ -a & 0 & 0 & & & \\ \hdashline 0 & 0 & 0 & & & \\ \hdashline & & & 0_3 & 0 & 0 \\ & & & 0 & 0_3 & 0 \\ & & & 0 & 0 & 0_3 \end{array}\right)$$

and the remaining 9 frontal slices look like

$$\left(\begin{array}{cc:c:ccc} 0 & 0 & 0 & & & \\ 0 & 0 & 0 & & \mathrm{G}_1^{(13)} & \\ \hdashline 0 & 0 & 0 & & 0_{1\times 9\times 9} & \\ \hdashline & & & 0_{3\times 3\times 9} & 0 & 0 \\ \mathrm{G}_1^{(23)} & & 0_{9\times 1\times 9} & 0 & 0_{3\times 3\times 9} & 0 \\ & & & 0 & 0 & 0_{3\times 3\times 9} \end{array}\right)$$

Since the a's only appear in positions with the same indices as they did in $\tilde{\mathrm{A}}$ (see Example 2.3.3), that portion is still alternating. For the G parts, note that the identity matrices in the first three frontal slices, when having their indices transposed, end up either in the $\mathrm{G}_1^{(13)}$ portion or the $\mathrm{G}_1^{(23)}$ portion, with appropriate signs.

Proof of correctness. Let $\mathbf{A}, \mathbf{B} \in \Lambda(n, \mathbb{F})^m$. Let $\hat{\mathrm{A}} = (\begin{bmatrix} \tilde{\mathrm{A}} & 0 \\ 0 & -\mathrm{G} \end{bmatrix}, \begin{bmatrix} 0 & \mathrm{G}^{(13)} \\ \mathrm{G}^{(23)} & 0 \end{bmatrix})$, $\hat{\mathrm{B}} = (\begin{bmatrix} \tilde{\mathrm{B}} & 0 \\ 0 & -\mathrm{G} \end{bmatrix}, \begin{bmatrix} 0 & \mathrm{G}^{(13)} \\ \mathrm{G}^{(23)} & 0 \end{bmatrix}) \in \mathrm{M}((n+m+(n+1)^2)\times(n+m+(n+1)^2)\times(n+m+(n+1)^2), \mathbb{F})$ be constructed from $\mathbf{A}$ and $\mathbf{B}$ using the procedure above, respectively.

We claim that $\mathbf{A}$ and $\mathbf{B}$ are pseudo-isometric if and only if $\hat{\mathrm{A}}$ and $\hat{\mathrm{B}}$ are equivalent as trilinear forms.

The only if direction. Suppose $P^{\mathrm{t}}\mathbf{A}P = \mathbf{B}^Q$ for some $P \in \mathrm{GL}(n, \mathbb{F})$ and $Q \in \mathrm{GL}(m, \mathbb{F})$.

We will construct a trilinear form equivalence from $\hat{\mathrm{A}}$ to $\hat{\mathrm{B}}$ of the form $S = \begin{bmatrix} P & 0 & 0 \\ 0 & Q^{-1} & 0 \\ 0 & 0 & R \end{bmatrix} \in \mathrm{GL}(n + m + (n+1)^2, \mathbb{F})$, where $R \in \mathrm{GL}((n+1)^2, \mathbb{F})$ is to be determined later on.

Recall that $\hat{\mathrm{A}} = (\begin{bmatrix} \tilde{\mathrm{A}} & 0 \\ 0 & -\mathrm{G} \end{bmatrix}, \begin{bmatrix} 0 & \mathrm{G}^{(13)} \\ \mathrm{G}^{(23)} & 0 \end{bmatrix})$, $\hat{\mathrm{B}} = (\begin{bmatrix} \tilde{\mathrm{B}} & 0 \\ 0 & -\mathrm{G} \end{bmatrix}, \begin{bmatrix} 0 & \mathrm{G}^{(13)} \\ \mathrm{G}^{(23)} & 0 \end{bmatrix})$. It can be verified that the action of S sends $\tilde{\mathrm{A}}$ to $\tilde{\mathrm{B}}$. It remains to show that, by choosing an appropriate R, the action of S also sends G to G.

Let G_1 be the first n frontal slices of G, and G_2 the last m frontal slices from G. Then the action of S sends G_1 to $R^t G_1^P R$, and G_2 to $R^t G_2^{Q^{-1}} R$. Since G_2 is all-zero, the action of S on G_2 results in an all-zero tensor, so we have $R^t G_2^{Q^{-1}} R = G_2$.

We then turn to G_1. For $i \in [n+1]$, consider the ith horizontal slice of G_1, which is of the form $H_i = \begin{bmatrix} 0 & B_{1,i} & B_{2,i} & \dots & B_{n,i} \end{bmatrix}$, where 0 denotes the $n \times (n+1)$ all-zero matrix, and $B_{j,i}$ is the $n \times (n+1)$ elementary matrix with the (j,i)th entry being 1, and other entries being 0. Note that those non-zero entries of H_i are in the $(k(n+1)+i)$th columns, for $k \in [n]$. Let $P^t = \begin{bmatrix} p_1 & \dots & p_n \end{bmatrix}$, where p_i is the ith column of P^t. Then P acts on H_i from the left, which yields $P^t H_i = \begin{bmatrix} 0 & P_{1,i} & \dots & P_{n,i} \end{bmatrix}$, where $P_{j,i}$ denotes the $n \times (n+1)$ matrix with the ith column being p_j, and the other columns being 0.

Let us first set $R = \begin{bmatrix} I_{n+1} & 0 \\ 0 & \hat{R} \end{bmatrix}$, where $\hat{R}$ is to be determined later on. Then the left action of R on G_1 preserves H_i through I_{n+1}. The right action of R on G_1 translates to the right action of $\hat{R}$ on H_i. To send $P^t H_i$ back to H_i, $\hat{R}$ needs to act on those $(k(n+1)+i)$th columns of H_i, $i \in [n+1]$, as P^{-1}. Note that for H_i and H_j, $i \neq j$, those columns with non-zero entries are disjoint. This gives $\hat{R}$ the freedom to handle different H_i's separately. In other words, $\hat{R}$ can be set as $P^{-1} \otimes I_{n+1}$. This ensures that for every H_i, $P^t H_i \hat{R} = H_i$. To summarize, we have $R^t G_1^P R = G_1$, and this concludes the proof for the only if direction.

The if direction. Suppose $\hat{A}$ and $\hat{B}$ are isomorphic as trilinear forms via $P \in \mathrm{GL}(n + m + (n+1)^2, \mathbb{F})$. Set $P = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix}$, where $P_{1,1}$ is of size $n \times n$, $P_{2,2}$ is of size $m \times m$, and $P_{3,3}$ is of size $(n+1)^2 \times (n+1)^2$. Consider the ranks of the frontal slices of $\hat{A}$.

- The ranks of the first n frontal slices are in $[2(n+1), 4n]$. This is because a frontal slice in this range consists of two copies of vertical slices of A (whose ranks are between $[0, n-1]$ due to the alternating condition), and one frontal slice of G (whose ranks are of $2(n+1)$).

- The ranks of the $n+1$ to $n+m$ frontal slices are in $[0, n]$. This is because a frontal slice in this range consists of only just one frontal slice of A.
- The ranks of the last $n(n+1)$ vertical slices are in $[0, 2n]$. This is because a frontal slice in this range consists of two copies of horizontal slices of G (whose ranks are either n or 1; see e.g. the form of H_i in the proof of the only if direction).

By the discussions above, we claim that P must be of the form $\begin{bmatrix} P_{1,1} & 0 & 0 \\ P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix}$. To see this, for the sake of contradiction, suppose there are non-zero entries in $P_{1,2}$ or $P_{1,3}$. Then a non-trivial linear combination of the first n frontal slices is added to one of the last $(m + (n+1)^2)$ frontal slices. This implies that for this slice, the lower-right $(n+1)^2 \times (n+1)^2$ submatrix is of the form $\begin{bmatrix} 0 & a_1 I_{n+1} & a_2 I_{n+1} & \dots & a_n I_{n+1} \\ -a_1 I_{n+1} & 0 & 0 & \dots & 0 \\ -a_2 I_{n+1} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_n I_{n+1} & 0 & 0 & \dots & 0 \end{bmatrix}$, where one of $a_i \in \mathbb{F}$ is non-zero. Then this slice is of rank $\geq 2(n+1)$, which is unchanged by left (resp. right) multiplying P^{t} (resp. P), so it cannot be equal to the corresponding slice of $\hat{\mathrm{B}}$ which is of rank $\leq 2n$. We then arrived at the desired contradiction.

Now consider the action of such P on the $n+1$ to $n+m$ frontal slices. Note that these slices are of the form $\begin{bmatrix} A_i & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. (Recall that the last m slices of G are all-zero matrices.) Then we have $\begin{bmatrix} P_{1,1}^{\mathrm{t}} & P_{2,1}^{\mathrm{t}} & P_{3,1}^{\mathrm{t}} \\ 0 & P_{2,2}^{\mathrm{t}} & P_{3,2}^{\mathrm{t}} \\ 0 & P_{2,3}^{\mathrm{t}} & P_{3,3}^{\mathrm{t}} \end{bmatrix} \begin{bmatrix} A_i & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_{1,1} & 0 & 0 \\ P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix} = \begin{bmatrix} P_{1,1}^{\mathrm{t}} A_i P_{1,1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. Since $P^{\mathrm{t}} \hat{\mathrm{A}}^P P = \hat{\mathrm{B}}$, we have $P^{\mathrm{t}} \hat{\mathrm{A}} P = \hat{\mathrm{B}}^{P^{-1}}$. Observe that for the upper-left $n \times n$ submatrices of the frontal slices of $\hat{\mathrm{B}}$, P^{-1} simply performs a linear

combination of B_i's. It follows that every $P_{1,1}^{\mathrm{t}} A_i P_{1,1}$ is in the linear span of B_i. Since we assumed $\dim(\langle A_i \rangle) = \dim(\langle B_i \rangle)$, we have that $\mathbf{A}$ and $\mathbf{B}$ are pseudo-isometric. This concludes the proof of Proposition 2.3.1. □

Chapter 3

The ALTEQ signature scheme

3.1 Chapter preliminaries

3.1.1 Defining ATFE and variants

Our proposed signature protocol ALTEQ 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 $(\phi \circ A)(u, v, w) := \phi(A^{\mathsf{t}}(u), A^{\mathsf{t}}(v), A^{\mathsf{t}}(w))$. 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 $\sum_{1 \le i < j < k \le n} c_{i,j,k} e_i^* \wedge e_j^* \wedge e_k^*$, where $c_{i,j,k} \in \mathbb{F}_q$, e_i is the ith standard basis vector, e_i^* is the linear form sending $u = (u_1, \ldots, u_n)^{\mathsf{t}} \in \mathbb{F}_q^n$ 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 $u = (u_1, \ldots, u_n)^{\mathrm{t}}$, $v = (v_1, \ldots, v_n)^{\mathrm{t}}$, $w = (w_1, \ldots, w_n)^{\mathrm{t}}$ 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 $(c_{i,j,k} : 1 \le i < j < k \le n)$, $c_{i,j,k} \in \mathbb{F}_q$, 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 3.1.1. 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 3.1.2. 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 3.1.3. 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 3.1.4. It is not known whether the search version of ATFE reduces to the decision version in polynomial time. In [GQ21a], it was shown that for some related problems, such as the quadratic form map isomorphism (cf. Definition 3.3.3), 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 generalizes 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.

3.1.2 Digital signatures

Definition 3.1.5. A signature scheme consists of a triplet of polynomial-time (possible probabilistic) algorithms (KeyGen, Sign, 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 C and an adversary $\mathcal{A}$ as the following.

- The challenger 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 C

  - 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 $\lambda^{-O(1)}$.

3.2 Signature schemes based on ATFE

Our scheme is inspired by the zero-knowledge protocol for graph isomorphisms by Goldreich, Micali and Wigderson (GMW) [GMW91]. At a high level, we will incorporate the ATFE to obtain a generalized GMW-like scheme and then apply the Fiat-Shamir transformation [FS86] to obtain a signature scheme. This basic scheme is described in Section 3.2.1. 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 3.2.1 for the detail.

3.2.1 The basic scheme

The original GMW protocol [GMW91] 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 [Pat96] and De Feo and Galbraith [FG19].

We present this slightly generalized scheme based on ATFE in Algorithms 1, 2, and 3. It involves five parameters: $n \in \mathbb{N}$ and a prime power q to specify $\mathrm{ATF}(n, q)$, the round number r, the security parameter λ, and the number of alternating trilinear forms in the public key C.

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 3.5.2.

Algorithm 1: Key generation.

Input: The variable number $n \in \mathbb{N}$, a prime power q, the alternating trilinear form number 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 $\phi_i \circ A_i = \phi_C$.

1 Randomly sample an alternating trilinear form $\phi_C : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$.
2 Randomly sample $C - 1$ invertible matrices, $A_1, \ldots, A_{C-1} \in \mathrm{GL}(n, q)$.
3 For every $i \in [C - 1]$, $\phi_i \leftarrow \phi_C \circ A_i$.
4 For every $i \in [C - 1]$, $A_i \leftarrow A_i^{-1}$.
5 $A_C \leftarrow I_n$.
6 **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. The message M. A hash function $H : \{0,1\}^* \to \{0,1\}^{2\lambda}$. An expander $\mathsf{Expand} : \{0,1\}^{2\lambda} \to \{a_i\}_{i \in [r]}$, where $a_i \in [C]$.

Output: The signature S on M.

1 **for** $i \in [r]$ **do**
2 Randomly sample $B_i \in \mathrm{GL}(n, q)$.
3 $\psi_i \leftarrow \phi_C \circ B_i$.
4 **end**
5 Compute $\mathsf{cha} = H(\mathrm{M}|\psi_1| \ldots |\psi_r) \in \{0,1\}^{\{2\lambda\}}$.
6 $(b_1, \ldots, b_r) \leftarrow \mathsf{Expand}(\mathsf{cha})$
7 **for** $i \in [r]$ **do**
8 $D_i \leftarrow A_{b_i} B_i$. ; // Note that $\phi_{b_i} \circ D_i = \psi_i$.
9 **end**
10 **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 ROM security of the digital signature scheme as follows. We also provide the QROM security in Section 4.4.

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 $H : \{0,1\}^* \to \{0,1\}^{2\lambda}$. An expander $\mathsf{Expand} : \{0,1\}^{2\lambda} \to \{a_i\}_{i \in [r]}$, where $a_i \in [C]$.

Output: "Yes" if S is a valid signature for M. "No" otherwise.

1 **for** $i \in [r]$ **do**
2 | Compute $\psi_i = \phi_{b_i} \circ D_i$.
3 **end**
4 Compute $\mathsf{cha}' = H(\mathrm{M}|\psi_1| \ldots |\psi_r) \in \{0,1\}^{2\lambda}$.
5 $(b'_1, \ldots, b'_r) \leftarrow \mathsf{Expand}(\mathsf{cha}')$
6 **if** *for every* $i \in [r]$, $b_i = b'_i$ **then**
7 | **return** *Yes*
8 **else**
9 | **return** *No*

Theorem 3.2.1. *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 modeled as random oracle, can break the EUF-CMA security, as described in Section 3.1.2, 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.

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 $\mathsf{cha}_1, \ldots, \mathsf{cha}_Q$ be random elements in $\{0,1\}^{2\lambda}$, 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 $(R, \phi_1, \ldots, \phi_C, \mathsf{cha}_1, \ldots, \mathsf{cha}_Q)$ 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 cha_j for $j \in [Q]$.

- Expand cha_j to the integers $b_{j1}, \ldots, b_{j,r} \in [C]$.
- For $i \in [r]$, choose randomly $D_i \leftarrow \mathrm{GL}(n, q)$ and set $\psi_i := \phi_{b_{ji}} \circ D_i$.
- Define $\mathsf{cha}_j := H(M|\psi_1| \ldots |\psi_r)$. If this value has already been defined then we pick another value 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 close 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 cha^* be the corresponding hash query of the adversary, i.e., cha^* 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 [BN06], $\mathcal{A}$ 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 $\mathsf{cha}^* \neq \mathsf{cha}'$, where cha' is programmed to be $H(M^*|\psi_1'| \cdots |\psi_r')$. Since $\mathsf{cha}^* \neq \mathsf{cha}'$, 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. □

3.3 Complexity and cryptography aspects of ATFE

3.3.1 ATFE in complexity theory

In Section 1.5, we mentioned the recent introduction of the Tensor Isomorphism-complete class (TI) in [GQ21b], which captures many isomorphism problems arising from multivariate crytography, machine learning, quantum information, and com-

puter algebra. In Chapter 2, ATFE was proved to be TI-complete. Among those TI-complete problems, the following algorithmic problems are of particular relevance to our discussion.

Definition 3.3.1. 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, characterizing equivalence classes of tripartite states under stochastic local operation and classical communication (SLOCC) [GQ21b].

Definition 3.3.2. The cubic form isomorphism problem (CFI) is the following.

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

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 [BFFP11] and theoretical computer science [AS05, AS06].

Definition 3.3.3. The quadratic form map isomorphism problem (QFMI) is the following.

Input Two tuples of quadratic forms $\mathbf{f} = (f_1, \ldots, f_m)$, $\mathbf{g} = (g_1, \ldots, g_m)$, where $f_i, g_j \in \mathbb{F}_q[x_1, \ldots, x_n]$ 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 [Pat96] and has been studied in several works including [FP06,BFV13,BFP15]. 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 3.3.4. 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 [O'B94, Wil09a, BMW17] and theoretical computer science [LQ17, BLQW20, GQ21b].

The following theorem is important for our understanding of ATFE.

Theorem 3.3.5 ([GQ21b, GQT21]). *The following problems are equivalent under polynomial-time reductions: ATFE, 3TI, CFI, QFMI, and pGpI.*

Theorem 3.3.5 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. In [GQT21], 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 $\phi \in \mathrm{ATF}(n, q)$, [1] where the constant hidden in the

[1] In [GQT21] an algorithm in such time was presented for CFI, but its algorithmic idea can be readily applied to ATFE.

big O is at least 4. Note that we don't include the average-case algorithm for ATFE in this thesis, because it is not useful in practice.

3.3.2 ATFE and cryptography based on group actions

Let G be a group and S a set. A group action is a function $\alpha : G \times S \longrightarrow S$ satisfying certain axioms. For the purpose of this thesis 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 [BY90], Couveignes [Cou06], and more recently in two papers [JQSY19, AFMP20]. We review this framework and explain the roles of the discrete logarithm problem and ATFE in this framework.

In [BY90], Brassard and Yung defined the group action α to be *one-way*, if there exists $s \in S$, such that $\alpha_s : G \longrightarrow S$, defined as $\alpha_s(g) = \alpha(g, s)$, is a one-way function. In [JQSY19], 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 [Cou06], shows how to interpret the discrete logarithm problem as a problem about group action.

Example 3.3.6. 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 $\alpha : \mathrm{Aut}(C_p) \times S \longrightarrow S$ 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 [Cou06], 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 3.3.6, its parallelization problem is hard on average is equivalent to the Computational Diffie-Hellman assumption.

Recently, the notion of pseudorandom group actions was independently introduced in [JQSY19] and [AFMP20].[2] 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 [JQSY19], it was observed that this assumption generalizes the Decisional Diffie-Hellman assumption. We reproduce this example here.

Example 3.3.7. Let C_p, $G = \mathrm{Aut}(C_p)$, and $S = C_p \setminus \{\mathrm{id}\}$ be from Example 3.3.6. 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 3.3.8. Consider the quadratic form map isomorphism problem (QFMI) from Definition 3.3.3, where $\mathrm{GL}(n, q) \times \mathrm{GL}(m, q)$ acts on tuples of quadratic forms

[2] In [AFMP20] this is called weak pseudorandom group actions.

$\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{F}_q[x_1, \ldots, x_n]$. 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 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 [Bou11]) suggests that Variant 1 is easier than Variant 2, 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 parallelization is hard. In [JQSY19, AFMP20], 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 3.3.1 (proposed in [JQSY19]) and the class group action underlying CSIDH [CLM+18] (proposed in [AFMP20]). 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 [AFMP20] require the group to be commutative.

Conjecture 3.3.9. 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 3.3.9.

- Several researchers have noted that the mathematics of alternating trilinear forms is "much harder" [Atk73], or "much more complicated (and interesting)" [DS14], 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 know, 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 [Hås90], 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 [JQSY19] supporting 3TI to be pseudorandom, including utilizing supergroups and invariant theory. These attack strategies work 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 3.3.9 as well.

3.4 Algorithms of the ATFE problem

We've shown how ATFE supports the EUF-CMA security of ALTEQ in ROM in Section 3.2.1. In Section 3.3.1, we gave pointers to the literature where the relations of many problems and the ATFE were discussed. So it remains to present the current status of the basic ATFE problem. We analyze known algorithms in Chapter 8, includ-

ing Gröbner basis attacks, graph-theoretic algorithms by Beullens [Beu23], as well as low-rank birthday attack. These attacks are accounted for in our parameter selection.

It should be noted that, because of the connections with many isomorphism problems, the algorithmic techniques for ATFE have been drawn from years of research experience of these computational areas for such problems.

3.5 Modern parameters and implementations.

The ALTEQ scheme implementation incorporates several measures to enhance the system performance. Some main points are as follows.

Unbalanced challenges. We incorporate the *unbalanced challenge* technique [FS86]. Briefly speaking, this means that in the GMW identification protocol, we set a fixed number of challenges to be some specific value. This is because when the challenge is of this value, the response is a random matrix expanded from a short seed, so sending this seed through reduces the communication (and thus the signature size). The cost is that more rounds are required, therefore increasing the sign and verification times. Specifically, we will sample r challenges $(b_1, \ldots, b_r) \in [C]^r$ with the property that $|\{i \in [r] \mid b_i = C\}| = r - K$. We denote K as the unbalanced challenge parameter.

Implementation considerations. The main algebraic operation is the group action computation, which relies on modular arithmetic. For modular arithmetic, we use a method for Pseudo-Mersenne numbers from [Cra92]. For group actions, we implement several optimizations, such as the tensorial viewpoint of alternating trilinear forms, and the use of decomposing an invertible matrix into a product of matrices in a special form.

Parameter choices. Let λ be the bit security level. To determine the choices of n and q (the ATFE parameters), we rely on two main approaches for solving ATFE: the Gröbner basis approach and the approach based on low-rank points. The Gröbner basis approach determines the vector space dimension n, and then the low-rank based

approach determines the field order q. The GMW-FS design parameters, namely the round number r and the form number C, and the unbalanced challenge parameter K, can be determined in a straightforward manner. There can be certain flexibility in getting some trade-offs between signature and public key sizes, as well as key generation, sign, and verify times.

3.5.1 Parameter choices

The choices of n. This is set up based on the direct Gröbner basis attack in Section 8.1. We compare the three modelings of polynomial systems below. We note that estimating the solving degrees of these systems is a major open problem. Lacking proper tools to understand them, we resort to the estimates of semi-regular systems [BFSY05]. We estimate that the direct Gröbner basis attack based on quadratic with inverse modellings, and the results are given in Table 3.3.

Practical evaluations of the three modelings. We carried out experiments for all the methods in Section 8.1 on Magma [BJP97].

All work for $n = 5$ on a laptop[3].

Modelling	Direct cubic	Quadratic with inverse	Quadratic dual
Time	< 0.01s	≈ 35s	≈ 11s
Step	4	15	13
Max degree	7	7	7
Memory	900MB	800 to 900MB	800 to 900 MB

Table 3.1: Performance of the three modelings for $n = 5$.

For $n = 6$, we put the experiments on a server[4].

[3]MacBook Pro, Apple M1 Pro chip, 32 GB memory.

[4]2x AMD EPYC 7532 2.40GHz 32 cores 256M L3 Cache (Max Turbo Freq. 3.33GHz), 1024GB 3200MHz ECC DDR4-RAM (Eight Channel).

Modelling	Direct cubic	Quadratic with inverse	Quadratic dual
Time	≈ 300s	between 79000s and 90000s	Could not finish after three weeks
Step	21	48	5 (stuck at)
Max degree	7	7	7 (stuck at)
Memory	4.2GB	167GB	170GB

Table 3.2: Performance of the three modellings for $n = 6$.

For $n = 7$, the direct cubic modeling failed after taking more than 300GB memory.

We computed the Hilbert series for the homogeneous parts of the three modelings, and they do not resemble generic polynomial systems with the same variable and equation numbers. To estimate the solving degrees and to investigate these modelings is an open problem.

Estimations based on semi-regular assumptions. We therefore adopt the following approach as a guide. We are aware that these systems are not homogeneous nor semi-regular, so this approach should not be applicable. But we resort to it due to the lack of appropriate tools at the moment.

First, we decide to follow the (*unrealistic*) assumption that these systems behave as semi-regular systems. Second, the regularity for cubic systems is usually much larger than quadratic ones, so for the sake of conservation, we drop the direct cubic modeling, despite that its performance is better than the other two. Third, we drop the quadratic dual modeling, because its performance at $n = 6$ is much worse than the other two.

This leaves us with the quadratic with inverse modeling. Following [YC04], we compute the regularity degrees d, use $\binom{2n^2+d}{d}$ as the Macaulay matrix sizes and $2 \cdot \binom{n}{2} + n$ as the density. Based on the formula $3 \cdot (\text{Macaulay-mat-size})^2 \cdot \text{density}$ as used in Rainbow [sCDK+21] and UOV [BCH+23], we have the following estimates for the number of *arithmetic operations*.

(1) $n = 13$, regularity $d = 11$, Macaulay-mat-size $\approx 2^{67}$, and arithmetic operations $\approx 2^{143}$.

(2) $n = 20$, regularity $d = 15$, Macaulay-mat-size $\approx 2^{104}$, and arithmetic operations $\approx 2^{219}$.

The choices of q. After selecting n, the choice of q is based on the low-rank birthday attack in Chapter 8. This relies on the rank statistics of n.

Let $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ be an alternating trilinear form. Let $\mathbb{P}(\mathbb{F}_q^n)$ be the projective space associated with $\mathbb{F}_q^n$, consisting of lines in $\mathbb{F}_q^n$. That is, for $v \in \mathbb{F}_q^n$, $v \neq 0$, we let $\hat{v} := \{u \in \mathbb{F}_q^n \mid u = \alpha \cdot v, \alpha \in \mathbb{F}_q\}$. For $\hat{v} \in \mathbb{P}(\mathbb{F}_q^n)$, let $\mathrm{rk}_\phi(\hat{v})$ be the rank of the bilinear form $\phi_{\hat{v}} := \phi(v, \cdot, \cdot)$. When it is clear from the context, we may just write as $\mathrm{rk}(\hat{v})$.

Based on Theorem 2 from [Beu23], the following data are most relevant to our choice.

(1) For $n = 13$, for a random ϕ, it is expected that $|\{\hat{v} \mid \mathrm{rk}_\phi(\hat{v}) = 8\}| \approx q^6$. It is also expected that $1/q^3$-fraction of ϕ has $\hat{v}$ such that $\mathrm{rk}_\phi(\hat{v}) = 6$.

(2) For $n = 20$, for a random ϕ, it is expected that $|\{\hat{v} \mid \mathrm{rk}_\phi(\hat{v}) = 14\}| \approx q^9$. It is also expected that $1/q^2$-fraction of ϕ has $\hat{v}$ such that $\mathrm{rk}_\phi(\hat{v}) = 12$.

The low-rank birthday algorithm described in Chapter 8 yields the following. Let $\text{minrank-cost}(n, k, r)$ denote the min-rank cost for sampling a rank-r matrix from the linear span of k $n \times n$ matrices.

(1) For $n = 13$, an algorithm with $O(q^3 \cdot \text{minrank-cost}(13, 7, 8) \cdot 13^6)$ arithmetic operations.

(2) For $n = 20$, an algorithm with $O(q^{4.5} \cdot \text{minrank-cost}(20, 11, 14) \cdot 20^6)$ arithmetic operations.

We use the algorithm[5] from [BBC+20] to estimate the min-rank cost as follows.

[5]We compared the estimates below with the estimates based on the analysis of the Kipnis–Shamir system [KS99] in [VBC+19], and found that the ones from [BBC+20] are lower.

parameter set	r	K	C	security level of ALTEQ (bit)
I	84	22	7	128.1
	16	14	458	130.6
III	201	28	7	192.0
	39	20	229	192.7

Table 3.4: The bit security of ALTEQ for the choices of C, r and K.

(1) minrank-cost$(13, 7, 8) \approx 2^{32}$.

(2) minrank-cost$(20, 11, 12) \approx 2^{57}$. Here we use the parameter $b = 4$ as in [BBC+20].

We now summarise the arithmetic complexities for the direct Göbner basis attack and the low-rank birthday attack for $n = 13$ and $n = 20$ with q being a 32-bit prime in Table 3.3.

	Quadratic with inverse GB, arithmetic	low-rank birthday, arithmetic
$n = 13, q = 2^{32} - 5$	$\approx 2^{143}$	$\approx 2^{128}$
$n = 20, q = 2^{32} - 5$	$\approx 2^{219}$	$\approx 2^{202}$

Table 3.3: Arithmetic complexities of the two attacks.

The above discussions are for numbers of arithmetic operations. These already suffice for levels I and III. To translate to bit complexities, we need to add the bit operation complexity for modular multiplications as $O(\log^2(q))$.

The choices of C, r, and K. We use the unbalanced challenge technique as mentioned above. This relies on three parameters, the round number r, the unbalanced parameter K, and the form number in each round C. To achieve the λ bit security, we require that $\binom{r}{K} \cdot C^K \geqslant 2^\lambda$. Table 3.4 illustrates the bit securities of our choices of r, K, and C.

3.5.2 Implementations

In this section, we provide the implementation details of ALTEQ. The code can be found at https://pqcalteq.github.io/. We explain some optimizations for modular arithmetic, group actions, and seed expansion.

Modular arithmetic. Operating on matrices and tensors requires multiple computations of a sum of products of elements over $\mathbb{F}_q$. Therefore, we will use only one single modulo i.e. $q = 2^{32} - 5$. This choice allows to use a large field without using multiprecision arithmetic. Consequently, each multiplication needs to be followed immediately by a modular reduction. Regarding modular addition, multiple operations can be done before a modular reduction. As q is a Pseudo-Mersenne number [Cra92], a modular reduction is done by a shift, an addition and multiplication by a constant. To guarantee that the result stays on 32 bits, a second round of modular reduction will need to be performed.

Representing invertible matrices and their actions. An invertible matrix is represented as a product of n invertible column matrices. Here, a *column matrix* is equal to the identity matrix for each coefficient but one column. Not all invertible matrices cannot be decomposed in such product (without the use of a permutation matrix), but the number of matrices not decomposable directly in such product of column matrices is negligible.

Once in the form of the product of n column matrices, a matrix can be applied to an alternating trilinear form in a simpler and faster way: each column matrix, one after the other, can be applied directly to the alternating trilinear without passing by a costly tensor form. Consequently, we obtain a reduction from $7/4 \cdot n^4$ to $1/2 \cdot n^4$ of the number of field multiplications required. This gain is especially evident in the verification process, as a majority of the cases are expanded from random seeds.

Finally, it is important to note that we can efficiently compute the matrix corresponding to the product of the column matrix by performing such a product itself. The product of a dense matrix by a column matrix will cost n^2 field multiplications. In this scheme, we will need to compute the product of $2n$ columns matrices. However, the first n column matrices product will cost less than n^2 fields multiplications because such product are with elements corresponding to zero and therefore does not need to be computed. The reason is that the identity matrix will still have $(n-1) \cdot (n-k)$

elements equal to zero after k products by a columns matrix. This is correct if columns are ordered as in our implementation.

Seed expansion. As we have multiple random objects to generate, we try to minimize the call to seed expander. To this end, for random matrices and random ATFs, we simply randomly generate a large number of values in $[0, 2^{32})$. The elements will discarded in the rare cases that is not falling in $[0, q)$ or if is equal to 0 in the case of element of the diagonal of a column matrix.

For generating the challenges, we need multiple values with different sizes: both for the challenge value different from C and to determine where the challenge is equal to C. This last step corresponds to picking K elements among r elements. It is important to note that to minimize the call to seed expander, the approach will be different if $r - K$ is smaller than K. For such cases, we pick $r - K$ elements among r elements for the same result. For all those reasons, when generating the challenges, we keep in a buffer each random bit generated by the seed expander to avoid unnecessary calls.

AVX2 acceleration. To fully utilize AVX acceleration, the representations of multiple ATFs have been intertwined: on the array representing ATF, the consecutive value does not correspond to the same ATF, but rather to the value having the same index in a different ATF. Concretely, the element corresponding to $ATF_r(i, j, k)$ is not followed by $ATF_r(i, j, k+1)$ but by $ATF_{r+1}(i, j, k)$. Consequently, when we need to compute the action of different matrices on multiple ATFs, this can be done in a vectorized manner.

Hashing function and seed expansion can also take advantage of AVX acceleration. For our symmetric needs, we borrow solutions from some previous submissions to NIST PQC standardization, such as Dilithium as well as from XKCP. The Dilithium team has already proposed an efficient and dedicated versions of AES utilizing AVX acceleration. Regarding Keccak, XKCP offers a version that is fully utilizing AVX as well. While these implementations have been slightly modified to fit our scheme, they should be fully credited to the Dilithium and XKCP teams.

Furthermore, while our implementation has been optimized and parameterized for AVX2, it will strongly take advantage of AVX512 as well.

Finally, our scheme could be accelerated even further by using multithreading. As it requires a dedicated implementation, we did not investigate such an option to focus principally on AVX2 acceleration.

Remark 3.5.1. Recently, two attacks against ALTEQ were released on the pqc forum. The first one is the forgery attack proposed by Saarinen [Saa23], which utilizes that the secret matrix does not do invertibility checking. The second is the multi-target attack proposed by Beullens. We have fixed the protocol by adding an invertibility check as well as doubling the seed size and adding salt. For the latest version of ALTEQ, please refer to https://pqcalteq.github.io/.

3.5.3 Performance analysis

We provide two sets of parameters for each security level I and III. The first set is called *Balanced*, and the second set is called *ShortSig*.

Key and signature sizes. In Table 3.5, we list the parameters for the balanced-ATFE-Sig for security levels I, III. Note that for level I, the public key+signature size is below 24KB. For level III, the public key+signature size is below 80KB.

Parameter set	Parameters (n, q, r, K, C)	Private key Size (Bytes)	Public key Size (Bytes)	Signature Size (Bytes)	Public key + signature Size (Bytes)
I	$(13, 2^{32} - 5, 84, 22, 7)$	16	8024	15896	23920
III	$(20, 2^{32} - 5, 201, 28, 7)$	24	31944	49000	80944

Table 3.5: Key and Signature Sizes for Balanced-ATFE-Sig

In Table 3.6, we list the parameters for the ShortSig-ATFE-Sig for security levels I, III. Note that for level I, the public key size is below 512KB and the signature size is below 10KB. For level III, the public key size is below 1MB and the signature size is below 32KB.

Parameter set	Parameters (n, q, r, K, C)	Private key Size (Bytes)	Public key Size (Bytes)	Signature Size (Bytes)
I	$(13, 2^{32} - 5, 16, 14, 458)$	16	523968	9528
III	$(20, 2^{32} - 5, 39, 20, 229)$	24	1044264	32504

Table 3.6: Key and Signature Sizes for ShortSig-ALTEQ

Performance. We test our codes on a machine with the following configurations.

- Processor: Intel Xeon E-2288G 3.7GHz 8 cores 16MB L3 Cache HT Enabled (Max Turbo Freq. 5.0GHz, Min 4.7GHz).
- Memory: 64GB.
- Operating system: Red Hat Enterprise Linux 8.6 (Ootpa).
- Compiler: gcc version 8.5.0 20210514 (Red Hat 8.5.0-10).

Our results are as follows. The numbers in the following Tables are averages over 1000 runs. We report the averages, and the medians are quite close to the averages.

parameter set		Key gen	Sign	Verify	Sign+verify
I	cycles	329285	2310789	1836795	4147584
	time (ms)	0.093	0.629	0.496	1.125
III	cycles	2121817	25965846	24075470	50041316
	time (ms)	0.582	6.986	6.483	13.469

Table 3.7: Performance of Balanced-ALTEQ.

parameter set		Key gen	Sign	Verify
I	cycles	7123223	686620	326242
	time (ms)	1.902	0.194	0.092
III	cycles	18339415	6346193	4851234
	time (ms)	5.152	1.705	1.304

Table 3.8: Performance of ShortSig-ALTEQ.

3.5.4 Comparison with other NIST submissions

In this section, we provide a comparison of ALTEQ with standardized protocol in Table 3.9, i.e. Dilithium [BDK+21], Falcon [FHK+20] and SPHINCS+ [ABWB+20].

We also include MEDS [CNP+23a] and LESS [BMPS20] in the Table 3.9 because, as mentioned in Section 1.8, the problems on which these two protocols are based are related to ATFE. As mentioned in Section 1.2, the choice of ATFE in post-quantum cryptography is backed by a strong limitation on known quantum algorithm techniques [HMR+10]. The speeds of our implementation, though still slower than lattice-based schemes, are acceptable in general. As for MEDS and LESS, ALTEQ's signature size is almost 1.5x theirs, but its key gen./signing/verification speed is much faster than theirs.

PQ security level	Algorithm	Public key Size (Bytes)	Signature Size (Bytes)	Key Gen. (cycles)	Signing (cycles)	Verification (cycles)
II	Dilithium2	1312	2420	7×10^4	2.5×10^5	7.2×10^4
I	Falcon512	897	666	19.9×10^9	3.8×10^8	8.2×10^7
I	SPHINCS+-128f	32	17088	1.8×10^7	4.6×10^8	2.8×10^7
I	LESS-1b	13600	8400	3.4×10^6	878.7×10^6	890.8×10^6
I	MEDS-9923	9923	9896	1.9×10^6	518.05×10^6	515.58×10^6
I	Balanced-ALTEQ	8024	15896	3.2×10^5	2.3×10^6	1.8×10^6

Table 3.9: Comparison with other submissions to the NIST

Part II

On digital signatures based on group action: QROM security and ring signatures

Chapter 4

Quantum Random Oracle Model (QROM) security

4.1 Chapter preliminaries

4.1.1 Notations

We collect some basic notation in this subsection. We use $\mathbb{F}_q$ to denote the finite field with q elements. The general linear group of degree n over $\mathbb{F}_q$ is denoted as $\mathrm{GL}(n, q)$. The base of the logarithm is 2 unless otherwise specified. For a finite set S, we use $s \xleftarrow{\$} S$ to denote that s is uniformly randomly sampled from S. Given a positive integer k, we denote by $[k]$ the set $\{1, \ldots, k\}$.

4.1.2 Σ-protocols

Let $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{W}$ be a binary relation, where $\mathcal{X}, \mathcal{W}, \mathcal{R}$ are recognizable finite sets. In other words, there is a polynomial time algorithm that can decide whether $(x, w) \in \mathcal{R}$ for $x \in \mathcal{X}$ and $w \in \mathcal{W}$. Given an instance generator Gen of a relation $\mathcal{R}$, the relation $\mathcal{R}$ is *hard* if for any poly-time quantum algorithm $\mathcal{A}$, the probability $\Pr[(x, w') \in \mathcal{R} \mid (x, w) \leftarrow \mathsf{Gen}(1^\lambda), w' \leftarrow \mathcal{A}(x)]$ is negligible.

Given a hard relation $\mathcal{R}$, the Σ-protocol for $\mathcal{R}$ is 3-move interactive protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ in which the prover $\mathcal{P}$ who has the witness w for the statement x tries to convince the verifier $\mathcal{V}$ that he possesses a valid witness w without revealing anything more than the fact that he knows w. Formally, Σ-protocol is defined as follows.

Definition 4.1.1. Let $\mathcal{R}$ be a hard binary relation. Let ComSet, ChSet, ResSet be the commitment space, challenge space and response space respectively. The Σ-protocol Σ for a relation $\mathcal{R}$ consists of three PPT algorithms $(\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2), \mathcal{V})$, where V is deterministic and we assume that $\mathcal{P}_1$ and $\mathcal{P}_2$ share the same state, working as the following:

- The prover $\mathcal{P}$ first computes a commitment $a \leftarrow \mathcal{P}_1(x, w)$ and sends a to the verifier $\mathcal{V}$.
- On input a commitment a, the $\mathcal{V}$ samples a random challenge c from the challenge space ChSet and sends to $\mathcal{P}$.
- $\mathcal{P}$ computes a response $r \leftarrow \mathcal{P}_2(x, w, a, c)$ and sends to the $\mathcal{V}$ who will run $\mathcal{V}(x, a, c, r)$ and outputs 1 if the transcript (a, c, r) is valid and 0 otherwise.

Identification from Σ-protocol. A Σ-protocol $(\mathcal{P}, \mathcal{V})$ with a key generation algorithm ID.Gen gives an identification scheme $(\mathsf{ID.Gen}, \mathcal{P}, \mathcal{V})$.

Completeness. A Σ-protocol is said to be complete if for all pair $(x, w) \in \mathcal{R}$, an honest prover $\mathcal{P}$ with $(\mathsf{pk}, \mathsf{sk})$, where $\mathsf{pk} := x$ and $\mathsf{sk} := w$, can always convince an honest verifier, i.e. $\Pr[\mathcal{V}(\mathsf{pk}, a, c, r) = 1 \mid a \leftarrow \mathcal{P}(\mathsf{sk}), c \in_R \mathsf{ChSet}, r \leftarrow \mathcal{P}_2(\mathsf{pubk}, \mathsf{sk}, a, c)] = 1$.

Post-Quantum 2-Soundness. We say a Σ-protocol has post-quantum 2-soundness, if for any λ and any poly-time quantum adversary $\mathcal{A}$, the following probability is negligible, taken over the randomness of $(x, w) \leftarrow \mathsf{Gen}(1^\lambda)$: $\Pr[\mathcal{V}(\mathsf{pk}, a, c, r) = 1 \wedge \mathcal{V}(\mathsf{pk}, a, c', r') = 1 \wedge c \neq c' \mid (a, c, r, c', r') \leftarrow \mathcal{A}(\mathsf{pk})] \leq \mathsf{negl}(\lambda)$.

Honest Verifier Zero Knowledge. A Σ-protocol has honest verifier zero knowledge (HVZK) if for all pairs $(x, w) \in \mathcal{R}$, there is a simulator $\mathcal{S}$ with only the statement x, can always compute a valid transcript (a, c, r), i.e. $\Pr[\mathcal{V}(\mathsf{pk}, a, c, r) = 1 \mid (a, c, r) \leftarrow \mathcal{S}(\mathsf{pk})] = 1$. Moreover, the output distribution of $\mathcal{S}$ on input (x, c) is equal to the distribution of those outputs generated via an honest execution conditioned on the verifier using c as the challenge.

Min-entropy. A Σ-protocol has α-bit min-entropy, if

$$\Pr_{(x,w)\in_R\mathcal{R}}[\text{min-entropy}(a|a \leftarrow \mathcal{P}_1(x, w)) \geq \alpha] \geq 1 - 2^{-\alpha}.$$

Commitment Recoverability. A Σ-protocol is commitment recoverable if given c and r, there is a unique a such that (a, c, r) is a valid transcript. Such a commitment may be publicly computed with the input (x, c, r).

Perfect Unique Response. A Σ-protocol has perfect unique response if for all pairs $(x, w) \in \mathcal{R}$, there is no two valid transcripts (a, c, r) and (a, c, r') of the same commitment a and challenge c but different responses $r \neq r'$, i.e. $\Pr[\mathcal{V}(x, a, c, r) = 1 \wedge \mathcal{V}(x, a, c, r') = 1 \wedge r \neq r'] = 0$.

Computationally Unique Response. A Σ-protocol has computationally unique response, if for any λ and any poly-time quantum adversary $\mathcal{A}$, the following probability is negligible, taken over the randomness of $(x, w) \leftarrow \mathsf{Gen}(1^\lambda)$:

$$\Pr[\mathcal{V}(x, a, c, r) = 1 \wedge \mathcal{V}(x, a, c, r') = 1 \wedge r \neq r' \mid (a, c, r, r') \leftarrow \mathcal{A}(x)] \leq \mathsf{negl}(\lambda).$$

4.1.3 Abstract group actions in cryptography

Let G be a group and S be a set. We use $*$ to denote the group multiplication. A group action is a function $\alpha : G \times S \to S$ satisfying certain natural axioms. There are several frameworks of group actions in cryptography [BY90, Cou06, JQSY19, AFMP20], which are mostly the same but can be different in some details. In this chapter, we use the following model.

Some notation. Let $\alpha : G \times S \to S$ be a group action. For $s \in S$, its *orbit* under α is $\mathcal{O}(s) := \{t \in S \mid \exists g \in G, \alpha(g,s) = t\}$, and its *stabilizer group* under α is $\mathrm{Stab}(s) = \{g \in G \mid \alpha(g,s) = s\}$. An element in $\mathrm{Stab}(s)$ is called an *automorphism* of s. By the orbit-stabilizer theorem, $|\mathcal{O}(s)| \cdot |\mathrm{Stab}(s)| = |G|$.

Computational assumptions. We first make the following computational assumptions for using a group action in algorithms.

(1) We work with group families $G = \{G_k\}_{k\in\mathbb{N}}$ and set families $S = \{S_k\}_{k\in\mathbb{N}}$.

(2) For a fixed k, G_k and S_k are finite, where $|S_k| = A_k$ and $|G_k| = B_k$, and $\log A_k$ and $\log B_k$ are upper bounded by some polynomial in k.

(3) The following tasks can be done in time polynomial in k: computing group product and inverse, deciding the equivalence of group elements, computing the group action function, and uniformly sampling group and set elements.

In the following, when k is clear from the context, we may just write G and S, and set $|S| = A$ and $|G| = B$.

We note that it is not necessary for a group action to satisfy all the above to be useful in cryptography. For example, the group action underlying CSIDH [CLM+18] cannot be efficiently computed for all group elements, though it can be modeled as a "restricted effective group action" as in [AFMP20].

Cryptographic assumptions. We now list the following assumptions for a group action to be useful in cryptography. Let $\alpha : G \times S \to S$ be a group action. Given $s \in S$, we shall often use the fact that we can sample from $\mathcal{O}(s)$ uniformly. This is because we can uniformly sample $g \in G$ and return $\alpha(g,s)$.

(1) One-way assumption: for $s \leftarrow_R S$ and $t \leftarrow_R \mathcal{O}(s)$, there is no probabilistic or quantum polynomial-time algorithm that returns g' such that $\alpha(g',s) = t$.

(2) Pseudorandom assumption: there is no probabilistic or quantum polynomial-time algorithm that can distinguish the following two distributions with non-negligible probability:

a) The random distribution: $(s, t) \in S \times S$ where $s, t \leftarrow_R S$.

b) The pseudorandom distribution: $(s, t) \in S \times S$ where $s \leftarrow_R S$, $t \leftarrow_R \mathcal{O}(s)$.

Those assumptions can be generalized to the following C-instance version.

Definition 4.1.2. Let $\alpha : G \times S \to S$ be a group action.

(1) We say that α satisfies the *C-one-way assumption*, if for $s_0 \leftarrow_R S$, given s_0 and $s_1, \ldots, s_{C-1} \leftarrow_R \mathcal{O}(s_0)$, there is no probabilistic or quantum polynomial-time algorithm that returns g', $i, j \in \{0, 1, \ldots, C-1\}$, $i \neq j$, such that $\alpha(g', s_i) = s_j$, with non-negligible probability.

(2) We say that α satisfies the *C-pseudorandom assumption*, if there is no probabilistic or quantum polynomial-time algorithm that can distinguish the following two distributions with non-negligible probability:

a) The random distribution: $(s_0, \ldots, s_{C-1}) \in S^C$ where $s_i \leftarrow_R S$.

b) The pseudorandom distribution: $(s_0, \ldots, s_{C-1}) \in S^C$ where $s_0 \leftarrow_R S$, and $s_1, \ldots, s_{C-1} \leftarrow_R \mathcal{O}(s_0)$.

Remark 4.1.3. These assumptions can also be restricted to the versions that work with a fixed s_0 rather than a random one. That is, in the above, replace $s_0 \leftarrow_R S$ with a fixed choice $s_0 \in S$. We shall call these C-one-way-$\mathcal{O}(s_0)$ and C-pseudorandom-$\mathcal{O}(s_0)$ assumptions, respectively.

The GMW-FS digital signature design. Let $\alpha : G \times S \to S$ be a group action. As mentioned in Section 1.7, we can obtain a digital signature by applying the Fiat-Shamir

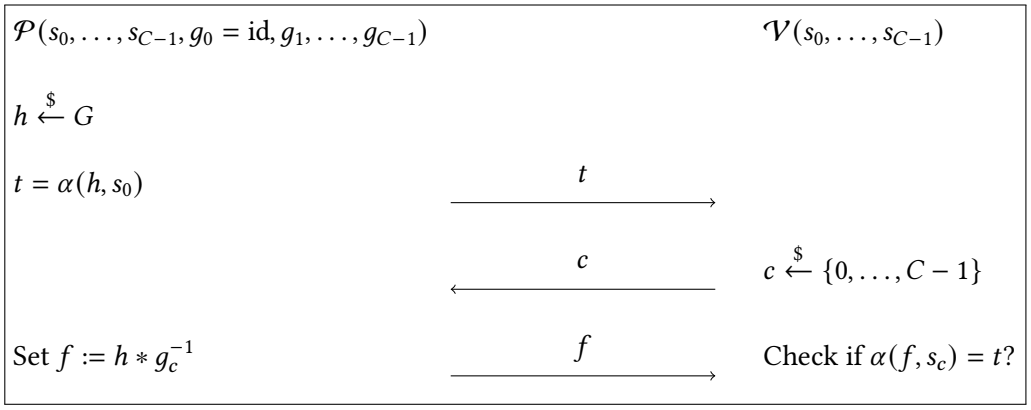

Figure 4.1: The $\alpha(\mathrm{G}, \mathrm{S})$-GMW protocol.

(FS) transformation to the Goldreich-Micali-Wigderson (GMW) zero-knowledge protocol instantiated with the group action α, assuming that the group action satisfies the C-one-way assumption. We call this digital signature the $\alpha(\mathrm{G}, \mathrm{S})$-GMW-FS scheme.

For our purposes in this part, the key is the GMW protocol instantiated with α with the C-one-way assumption. This protocol is easily interpreted as an identification protocol, and we shall refer it as the $\alpha(\mathrm{G}, \mathrm{S})$-GMW protocol. Therefore, we describe the $\alpha(\mathrm{G}, \mathrm{S})$-GMW protocol in detail.

In the $\alpha(\mathrm{G}, \mathrm{S})$-GMW protocol, the public key consists of set elements $s_0, \ldots, s_{C-1}$ such that $s_0 \leftarrow_R S$, and $s_1, \ldots, s_{C-1} \leftarrow_R \mathcal{O}(s_0)$. The private keys consists of $g_0 = \mathrm{id}, g_1, \ldots, g_{C-1}$ such that $\alpha(g_i, s_0) = s_i$. In this protocol, the goal of the prover is to convince the verifier that, for every $i \neq j$, the prover knows some h such that $\alpha(h, s_i) = s_j$.

Define the relation $R := \{x = \{s_0, \ldots, s_{C-1}\}, w = \{g_1, \ldots, g_{C-1}\} \mid x \subseteq S, w \subseteq G, \alpha(g_i, s_1) = s_i, \forall i \in \{1, \ldots, C-1\}\}$. The protocol is described in Figure 4.1, which will be repeated several times to attain the required security level.

The $\alpha(\mathrm{G}, \mathrm{S})$-GMW-FS-$\mathcal{O}(s)$ scheme. In Section 4.2, we will need a variant of the $\alpha(\mathrm{G}, \mathrm{S})$-GMW-FS-$\mathcal{O}(s)$ scheme, following Remark 4.1.3. Briefly speaking, this variant restricts to an orbit of some specific $s \in S$ instead of working in the orbit of a random $s \leftarrow_R S$. We call such a scheme the $\alpha(\mathrm{G}, \mathrm{S})$-GMW-FS-$\mathcal{O}(s)$ scheme.

4.1.4 Properties of the Σ-protocol based on abstract group actions

Completeness. It is clear that the honest prover with the statement and witness (x, w) following the $\alpha(\mathrm{G}, \mathrm{S})$-GMW protocol can always convince the honest verifiers.

Post-Quantum 2-Soundness. If there is a poly-time quantum adversary $\mathcal{A}$ with statement $x = \{s_0, \ldots, s_{C-1}\}$ who can compute two valid transcripts (t, c, h) and (t, c', h') where $c \neq c'$. Since $\alpha(h, s_c) = t$ and $\alpha(h', s_{c'}) = t$, the adversary $\mathcal{A}$ can get $f = h^{-1} * h'$ such that $s_c = \alpha(f, s_{c'})$, which is contradicted to the group action one-way assumption.

HVZK. Given a statement $x = \{s_0, \ldots, s_{C-1}\}$, there is a simulator $\mathcal{S}$ first sampling $c \in_R \{0, \ldots, C-1\}$ and $h \in_R G$ and then computing $t = \alpha(h, s_c)$. It follows that (t, c, h) is a valid transcript. Then the distributions of h and c are uniform, and $t = \alpha(h, s_c)$ is uniformly from the orbit where statement x is in. The distribution of $(t, c, h) \leftarrow \mathcal{S}(x)$ is equal to the distribution of real transcripts since both are uniform distributions on commitments, challenges, and responses.

Min-Entropy. Since commitment t is uniformly taken from the orbit O which elements of the statement $x = \{s_0, \ldots, s_{C-1}\}$ belong to, the $\alpha(\mathrm{G}, \mathrm{S})$-GMW protocol has α-bit min-entropy with $\alpha = \log_2(|O|)$ and $|O|$ is the size of orbit O.

Remark 4.1.4. By the orbit-stabiliser theorem, for an alternating trilinear form ϕ over $\mathbb{F}_q^n$, we have $|O(\phi)| = |\mathrm{GL}(n, q)|/|\mathrm{Aut}(\phi)|$. In Section 4.4, some results on the automorphism group orders, and therefore orbit sizes, of random alternating trilinear forms will be presented.

Commitment Recoverable. The $\alpha(\mathrm{G},\mathrm{S})$-GMW protocol is commitment recoverable. In fact, given a challenge c and a response h, there is only one commitment t computed by $t = \alpha(h, s_c)$.

4.1.5 Some candidates of group actions for the GMW-FS design

The group action underlying ALTEQ. We recall the definition of the group action on ATF. 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. We use $\mathrm{ATF}(n,q)$ to denote the set of all alternating trilinear forms defined over $\mathbb{F}_q^n$. 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 group action underlying LESS [BMPS20]. For $1 \le d \le n$, let $\mathrm{M}(d \times n, \mathbb{F}_q)$ be the linear space of $d \times n$ matrices over $\mathbb{F}_q$. Let $\mathrm{Mon}(n,q)$ be the group of $n \times n$ monomial matrices over $\mathbb{F}_q$. The group $G = \mathrm{GL}(n,q) \times \mathrm{Mon}(n,q)$, the set $S = \mathrm{M}(d \times n, \mathbb{F}_q)$, and the action is defined as $(A,C) \in \mathrm{GL}(n,q) \times \mathrm{Mon}(n,q)$ sending $B \in \mathrm{M}(d \times n, q)$ to ABC^t.

The group action underlying MEDS [CNP+23b]. Let $n_1, n_2, n_3 \in \mathbb{N}$. The set S is $\mathbb{F}_q^{n_1} \otimes \mathbb{F}_q^{n_2} \otimes \mathbb{F}_q^{n_3}$. The group $G = \mathrm{GL}(n_1,q) \times \mathrm{GL}(n_2,q) \times \mathrm{GL}(n_3,q)$. The action is defined as $(A_1, A_2, A_3) \in G$ sending $u_1 \otimes u_2 \otimes u_3$ to $A_1(u_1) \otimes A_2(u_2) \otimes A_3(u_3)$, and then linearly extending this to the whole $\mathbb{F}_q^{n_1} \otimes \mathbb{F}_q^{n_2} \otimes \mathbb{F}_q^{n_3}$.

The class group action such as CSIDH [CLM+18] (for SeaSign [FG19] and CSI-FiSh [BKV19]). Let E be an elliptic curve over $\mathbb{F}_p$, and let $O := \mathrm{End}_{\mathbb{F}_p}(E)$. The ideal class group $\mathrm{Cl}(O)$ acts on the set of $\mathbb{F}_p$-isomorphism classes of elliptic curves with $\mathbb{F}_p$-rational endomorphism ring O via a natural action. For details we refer the reader to [FG19, BKV19, BFGP23]. Note that this action does not satisfy all the properties in Section 4.1.3; see [AFMP20].

Further group actions in cryptography. We note that more isomorphism problems and group actions have been proposed for cryptographic uses, such as lattice isomorphism [DvW22] and knot equivalence [FGH+12]. While these are interesting, we did not discuss these here, because they have not been used with the GMW-FS design which is the focus of this part.

4.2 QROM security via perfect unique responses

In this section, we show that the $\alpha(\mathrm{G},\mathrm{S})$-GMW-FS scheme is secure in the quantum random oracle model (QROM) subject to a certain condition on the automorphism group of the alternating trilinear form in use.

This section is organized as follows. In Section 4.2.1, we translate perfect and computational unique response properties of the $\alpha(\mathrm{G},\mathrm{S})$-GMW protocol to certain properties about stabilizer groups. In Section 4.2.2, we formally state QROM security of the $\alpha(\mathrm{G},\mathrm{S})$-GMW-FS-$\mathcal{O}(s_0)$ scheme in Theorem 4.2.7, with proof sketches in Section 4.2.3.

4.2.1 Perfect and computationally unique responses of the $\alpha(\mathrm{G},\mathrm{S})$-GMW protocol

We require some extra properties such that the $\alpha(\mathrm{G},\mathrm{S})$-GMW or $\alpha(\mathrm{G},\mathrm{S})$-GMW-$\mathcal{O}(s_0)$ protocols meet the *perfect unique response* and *computationally unique response* properties, as recalled in Section 4.1.2.

Lemma 4.2.1 (Perfect Unique Response)**.** *The $\alpha(\mathrm{G},\mathrm{S})$-GMW-$\mathcal{O}(s_0)$ protocol supports perfect unique response if and only if* $\mathrm{Stab}(s_0)$ *is trivial.*

Proof. Assume that $\mathrm{Stab}(s_0)$ is trivial. If there are two valid transcripts (t, c, g_1) and (t, c, g_2) for the protocol in Figure 4.1. Then we have $\alpha(g_1, t) = \alpha(g_2, t)$. It implies that $g_2 * g_1^{-1} \in \mathrm{Stab}(s_0)$ and thus $g_1 = g_2$.

Now assume that the $\alpha(\mathrm{G},\mathrm{S})$-GMW-$\mathcal{O}(s_0)$ protocol satisfies the perfect unique response property. If $\mathrm{Stab}(s_0)$ is non-trivial, i.e., there exists a group element $h \neq \mathrm{id}$ such that $\alpha(h, s_0) = s_0$. Therefore, all the elements in $\{s_0, \ldots, s_{C-1}\}$ satisfy $\alpha(h, s_i) = s_i$. It follows that for the statement $\{s_0, \ldots, s_{C-1}\}$, any commitments $t \in S$, and any challenge $c \in \{0, 1, \ldots, C-1\}$, there are two different responses $g \in G$ and $h * g \in G$ such that (t, c, g) and $(t, c, h * g)$ are valid transcripts, which is a contradiction. □

Remark 4.2.2. For the $\alpha(\mathrm{G},\mathrm{S})$-GMW, since s_0 is not fixed, in some cases, we can only say that the stabilizer group of a random $s_0 \leftarrow_R S$ is trivial with high probability. Such a property is known as the statistical unique response property. However, it is not known if statistical unique response is enough to prove the quantum proof of knowledge.

To illustrate the relation between the computationally unique response and group actions, we define the following algorithm problem.

Definition 4.2.3. The $\alpha(\mathrm{G},\mathrm{S})$-stabilizer problem is the following.

Input: An element $s \in_R S$.

Output: Some $g \in G, g \neq \mathrm{id}$ such that $s = \alpha(g, s)$.

The $\alpha(\mathrm{G},\mathrm{S})$-stabilizer problem is also known as the automorphism group problem in the literature (see e.g. the graph automorphism problem [KST93]).

Lemma 4.2.4 (Computationally Unique Response)**.** *The $\alpha(\mathrm{G},\mathrm{S})$-GMW protocol in Figure 4.1 supports computationally unique response if and only if no poly-time quantum algorithm can solve the $\alpha(\mathrm{G},\mathrm{S})$-stabilizer problem in Definition 4.2.3 with a non-negligible probability.*

Proof. Assume that the Σ-protocol supports computationally unique response. If there is a polynomial-time quantum adversary $\mathcal{A}$ such that for any statement $x = \{s_0, \ldots, s_{C-1}\} \subseteq S$, it can compute two valid transcripts (t, c, g_1) and (t, c, g_2), where

$g_1 \neq g_2$, with a non-negligible probability. Then there is an algorithm $\mathcal{A}_1$ using $\mathcal{A}$ as subroutine such that for any $c \in \{0, 1, \ldots, C-1\}$, it can produce an $h = g_2 * g_1^{-1}$ such that $\alpha(h, s_c) = s_c$ with a non-negligible probability.

Assume there is a polynomial-time quantum algorithm $\mathcal{A}_1$ such that, for any $s \in S$, it produces a stabilizer element h such that $\alpha(h, s) = s$ with a non-negligible probability. By the HVZK property, there exists a simulator $\mathcal{S}$ such that, for any $x = \{s_0, \ldots, s_{C-1}\} \subseteq S$, it produces a valid transcript (t, c, g). Then there is an adversary $\mathcal{A}$ using $\mathcal{A}_1$ and $\mathcal{S}$ as subroutines such that it firstly computes a valid transcript (t, c, g) by $\mathcal{S}$, and then computes h such that $\alpha(h, s_c) = s_c$ by $\mathcal{A}_1$. Thus, for any statement $\{s_0, \ldots, s_{C-1}\}$, $\mathcal{A}$ computes two transcripts (t, c, g) and $(t, c, h * g)$ with a non-negligible probability. □

Remark 4.2.5. For a fixed $s_0 \in S$, we can define the $\alpha(\mathrm{G}, \mathrm{S})$-stabilizer-$\mathcal{O}(s_0)$ problem by restricting the input to $s \in_R \mathcal{O}(s_0)$. Then the above proof can be applied to show the same result for $\alpha(\mathrm{G}, \mathrm{S})$-GMW-$\mathcal{O}(s_0)$.

Based on the above, we define the following properties of group actions.

Definition 4.2.6. Let $\alpha : G \times S \to S$ be a group action.

(1) We say that α satisfies the (statistical) trivial stabilizer assumption, if for a random $s \in S$, $\mathrm{Stab}(s)$ is trivial.

(2) We say that α satisfies the non-trivial automorphism hardness assumption, if no probabilistic or quantum polynomial-time algorithm can solve the $\alpha(\mathrm{G}, \mathrm{S})$-stabilizer problem with non-negligible probability.

4.2.2 QROM security via perfect unique response

Lemma 4.2.1 interprets the perfect unique response property as a property of group actions. Based on this, it is straightforward to adapt the results in [LZ19] to give a security proof in QROM for $\alpha(\mathrm{G}, \mathrm{S})$-GMW-FS-$\mathcal{O}(s_0)$ signature scheme assuming the stabilizer group being trivial.

Theorem 4.2.7. *Suppose $s_0 \in S$ satisfies that* $\mathrm{Stab}(s_0)$ *is trivial, and assume the C-one-way-$O(s_0)$ is hard. The $\alpha(\mathrm{G},\mathrm{S})$-GMW-FS-$O(s_0)$ signature based on the t repetitions of $\alpha(\mathrm{G},\mathrm{S})$-GMW-$O(s_0)$ protocol has existential unforgeability under chosen-message attack (EUF-CMA) security. More specifically, for any polynomial-time quantum adversary $\mathcal{A}$ querying the quantum random oracle Q_H times against EUF-CMA security of $\alpha(\mathrm{G},\mathrm{S})$-GMW-FS-$O(s_0)$ signature, there is a quantum adversary $\mathcal{B}$ for C-one-way-$O(s_0)$ problem such that,*

$$\mathsf{Adv}_{\mathcal{A}}^{\alpha(\mathrm{G},\mathrm{S})\text{-}EUF\text{-}CMA} \leq O\left(Q_H^9 \cdot \left(\mathsf{Adv}_{\mathcal{B}}^{C\text{-}one\text{-}way\text{-}O(s_0)}\right)^{\frac{1}{3}}\right).$$

Remark 4.2.8. The EUF-CMA security in QROM here can be strengthened to the sEUF-CMA security by assuming the computationally unique response property [KLS18, Theorem 3.2]. Since we assume that the stabilizer group is trivial (perfect unique response) which implies the computationally unique response, $\alpha(\mathrm{G},\mathrm{S})$-GMW-FS-$O(s_0)$ signature here is sEUF-CMA secure.

Remark 4.2.9. The ATFE instantiation in Section 4.4 provides meaningful realization of Theorem 4.2.7 in a concrete group action setting. In fact, in Section 4.4, we experimentally verify that for a random alternating trilinear form, its stabilizer group is trivial, hence supporting the security of ALTEQ in the QROM model; see Section 4.4 for more detail.

4.2.3 Proof of Theorem 4.2.7

To prove Theorem 4.2.7 we first need some preparations.

Post-Quantum ID soundness of $\alpha(\mathrm{G},\mathrm{S})$-GMW-$O(s_0)$ Σ-protocol. When a Σ-protocol is for identification, we need a definition of ID soundness to protect against adversaries with eavesdropping attacks.

Definition 4.2.10. A Σ-protocol has *post-quantum ID soundness* if for any $(x,w) \in R$, every adversary $\mathcal{A}^{O_{\mathcal{P},\mathcal{V}}} = \left(\mathcal{A}_0^{O_{\mathcal{P},\mathcal{V}}}, \mathcal{A}_1^{O_{\mathcal{P},\mathcal{V}}}\right)$ with only the pk and polynomial times

of queries to the valid transcripts generated with an honest prover $\mathcal{P}$ with pk and sk and an honest verifier $\mathcal{V}$ with pk can convince an honest verifier $\mathcal{V}$ with a negligible probability, i.e., the probability

$$\Pr\left[\mathcal{V}.\mathsf{Ver}(\mathsf{pk}, a, c, r) = 1 \mid a \leftarrow \mathcal{A}_0^{O_{\mathcal{P},\mathcal{V}}}(\mathsf{pk}) \wedge c \leftarrow_R \{0,1\}^\lambda \wedge r \leftarrow \mathcal{A}_1^{O_{\mathcal{P},\mathcal{V}}}(\mathsf{pk}, a, c)\right].$$

is negligible.

Liu and Zhandry show that post-quantum identification soundness can be satisfied if a Σ-protocol has the weakly collapsing property and some extra properties [LZ19, Theorem 1]. Since the perfect unique response is a stronger property than the weakly collapsing property, we can state the result in [LZ19] as follows.

Theorem 4.2.11 ([LZ19]). *If a Σ-protocol with an exponentially large challenge space has completeness, post-quantum 2-soundness, HVZK, and perfect unique response, it is a Σ-protocol with post-quantum ID soundness that for any polynomial-time quantum adversary $\mathcal{A}$ against post-quantum ID soundness, there is a quantum adversary $\mathcal{B}$ for 2-soundness such that,*

$$\mathsf{Adv}_{\mathcal{A}}^{\textit{ID-sound}} \leq O\left(\left(\mathsf{Adv}_{\mathcal{B}}^{\textit{2-sound}}\right)^{\frac{1}{3}}\right).$$

Corollary 4.2.12. *Let $\alpha : G \times S \to S$ be a group action. Suppose we have some $s_0 \in S$ such that $\mathrm{Stab}(s_0)$ is trivial. The t repetitions of $\alpha(\mathrm{G},\mathrm{S})$-GMW-$O(s_0)$ Σ-protocol in Figure 4.1 is a Σ-protocol with post-quantum ID soundness that for any polynomial-time quantum adversary $\mathcal{A}$ against post-quantum ID soundness, there is a quantum adversary $\mathcal{B}$ for C-one-way-$O(s_0)$ problem such that,*

$$\mathsf{Adv}_{\mathcal{A}}^{\alpha(\mathrm{G},\mathrm{S})-\textit{ID}} \leq O\left(\left(\mathsf{Adv}_{\mathcal{B}}^{\textit{C-one-way-}O(s_0)}\right)^{\frac{1}{3}}\right).$$

Proof. As $\mathrm{Stab}(s_0)$ is trivial, by Lemma 4.2.1, the Σ-protocol in Figure 4.1 has perfect unique response. It also satisfies completeness, 2-soundness, and HVZK. Since the t repetitions of Σ-protocol in Figure 4.1 has an exponentially large challenge space, we can conclude the proof by Theorem 4.2.11. □

Security of $\alpha(\mathrm{G},\mathrm{S})$-GMW-FS-$\mathcal{O}(s_0)$ signature. Liu and Zhandry [LZ19, Theorem 11] showed that the signature security can be reduced to the underlying Σ-protocol with post-quantum ID soundness through a variant of Zhandry's compressed oracle model [Zha19]. Since min-entropy $\alpha = \Omega(\lambda)$ implies that the Σ-protocol has unpredictable commitment, we can substitute unpredictable commitment with $\Omega(n)$ bits min-entropy to have the following theorem.

Theorem 4.2.13 ([LZ19], Theorem 1)**.** *If a Σ-protocol has post-quantum ID soundness and $\Omega(n)$ bits min-entropy, the Fiat-Shamir transformation can produce a signature scheme with EUF-CMA security that for any polynomial-time quantum adversary $\mathcal{A}$ querying the quantum random oracle Q_H times against EUF-CMA security, there is a quantum adversary $\mathcal{B}$ against ID-soundness of the underlying protocol such that,*

$$\mathsf{Adv}^{\textit{EUF-CMA}}_{\mathcal{A}} \leq O\left(Q_H^9 \cdot \mathsf{Adv}^{\textit{ID-sound}}_{\mathcal{B}}\right).$$

Corollary 4.2.14. *If the t repetitions of $\alpha(\mathrm{G},\mathrm{S})$-GMW-$\mathcal{O}(s_0)$ protocol showed in Figure 4.1 has post-quantum ID soundness, then the corresponding Fiat-Shamir signature has EUF-CMA security that for any polynomial-time quantum adversary $\mathcal{A}$ querying the quantum random oracle Q_H times against EUF-CMA security of $\alpha(\mathrm{G},\mathrm{S})$-GMW-FS-$\mathcal{O}(s_0)$ signature, there are quantum adversary $\mathcal{B}$ against ID-soundness of $\alpha(\mathrm{G},\mathrm{S})$-GMW-$\mathcal{O}(s_0)$ protocol such that,*

$$\mathsf{Adv}^{\alpha(\mathrm{G},\mathrm{S})-\textit{EUF-CMA}}_{\mathcal{A}} \leq O\left(Q_H^9 \cdot \mathsf{Adv}^{\alpha(\mathrm{G},\mathrm{S})-\textit{ID}}_{\mathcal{B}}\right).$$

Proof. Assume the t repetitions of Σ-protocol showed in Figure 4.1 has post-quantum ID soundness. We proved that it has $\log_2(|\mathcal{O}(s_0)|)$ bits min-entropy, and $|\mathcal{O}(s_0)| = 2^{\Omega(\lambda)}$. Now we complete the proof utilizing the result of Theorem 4.2.13. □

We are now ready to prove Theorem 4.2.7.

Proof of Theorem 4.2.7. By Corollary 4.2.12, we have a Σ-protocol with post-quantum ID soundness. Then the EUF-CMA security can be achieved by Corollary 4.2.14.□

4.3 QROM security via lossy schemes

4.3.1 Definitions and previous results

In this section, we recall the definition of lossy identification protocol [AFLT12,EKP20] and a security result of its associated Fiat-Shamir signature in QROM from [KLS18].

Definition 4.3.1. An identification protocol ID is called *lossy*, denoted by $\mathsf{ID}_{\mathsf{ls}}$, if it has one additional PPT algorithm LossyGen, called the lossy key generation that on inputs the security parameter outputs a lossy verification key pubk. To be more precise, $\mathsf{LossyGen}(1^\lambda)$ generates $x_{\mathsf{ls}} \leftarrow \mathsf{LossyGen}(1^\lambda)$ such that there are no $w \in \mathcal{W}$ satisfying $(x_{\mathsf{ls}}, w) \in \mathcal{R}$.

A lossy identification protocol is required to satisfy the following additional properties.

Indistinguishability of lossy statements. It is required that the lossy statements generated by $\mathsf{LossyGen}(1^\lambda)$ is indistinguishable with ones generated by $\mathsf{Gen}(1^\lambda)$, i.e., . for any PPT (or quantum PT) adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ against the indistinguishability of lossy statements

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ls}}_{\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(x_{\mathsf{ls}} = 1)|x_{\mathsf{ls}} \leftarrow \mathsf{LossyGen}(1^\lambda)] \\ - \Pr[\mathcal{A}(x) = 1|(x, w) \leftarrow \mathsf{Gen}(1^\lambda)]\end{aligned}$$

is negligible.

Statistical lossy soundness. Consider following experiment $\mathsf{Exp}^{\mathsf{ls}}_{\mathsf{ID},\mathcal{A}}(\lambda)$ between an adversary $\mathcal{A}$ and a challenger.

- The challenger runs $x_{\mathsf{ls}} \leftarrow \mathsf{LossyGen}(1^\lambda)$ and provides x_{ls} to the adversary $\mathcal{A}$.
- On input x_{ls}, the adversary $\mathcal{A}$ selects a commitment a and sends it to the challenger who responds with a random challenge c.
- On input (a, c), the adversary $\mathcal{A}$ outputs a response r.

- Return 1 if (a, c, r) is a valid transcript for x_{ls}, and 0 otherwise.

We say that the lossy identification protocol $\mathsf{ID}_{\mathsf{ls}}$ is ϵ_{ls}-lossy sound if for any unbounded (possibly quantum) adversary $\mathcal{A}$, the probability of winning the experiment $\mathsf{Exp}^{\mathsf{ls}}_{\mathsf{ID},\mathcal{A}}(\lambda)$ is less than ϵ_{ls}, i.e.,

$$\Pr[\mathsf{Exp}^{\mathsf{ls}}_{\mathsf{ID},\mathcal{A}}(\lambda) = 1] \leq \epsilon_{\mathsf{ls}}.$$

Fiat-Shamir transformation applied to a lossy identification protocol yields a tightly secure signature in QROM [KLS18, LZ19, DFMS19].

Theorem 4.3.2 ([KLS18, Theorem 3.1]). *Assume that the identification protocol ID is lossy, perfect HVZK, has α bits of min-entropy, and it is ϵ_{ls}-lossily sound. Then the signature scheme FS[ID] obtained from applying the Fiat-Shamir transformation to ID is such that for any quantum adversary $\mathcal{A}$ against the sEUF-CMA security that issues at most Q_H queries to the quantum random oracle, there exist a quantum adversary $\mathcal{B}$ against the lossiness and C against the computation unique response such that*

$$\mathsf{Adv}^{\mathit{sEUF\text{-}CMA}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{ls}}_{\mathcal{B}}(\lambda) + 2^{-\alpha+1} + 8(Q_H+1)^2 \cdot \epsilon_{\mathsf{ls}} + \mathsf{Adv}^{\mathit{CUR}}_{C}(\lambda),$$

and $\mathsf{Time}(\mathcal{B}) = \mathsf{Time}(C) = \mathsf{Time}(\mathcal{D}) = \mathsf{Time}(\mathcal{A}) + Q_H \cong \mathsf{Time}(\mathcal{A})$.

In the classical setting, we can replace $8(Q_H+1)^2$ *by* (Q_H+1).

4.3.2 Lossy identification protocol from abstract group actions

In this section, we define a lossy identification protocol based on the K-pseudorandom assumption in Definition 4.1.2. The underlying sigma protocol is the $\alpha(\mathsf{G},\mathsf{S})$-GMW protocol in Figure 4.1. Here, we consider a relation $\mathcal{R}$ consisting of statement-witness pairs (x, w) with $x = \{s_0, s_1, \ldots, s_{C-1}\} \subseteq S$ and $w = \{g_1, \ldots, g_{C-1}\} \subseteq G$, where $\alpha(g_i, s_0) = s_i$ for each $i \in [C-1]$.

The lossy identification scheme for the relation $\mathcal{R}$ defined as above with challenge space $\{0, 1, \cdots, C-1\}$ consists of five algorithms $(\mathsf{IGen}, \mathsf{LossyGen}, \mathcal{P}_1, \mathcal{P}_2, \mathcal{V})$ as follows. Note that the new addition is the LossyGen algorithm.

- Algorithm IGen randomly samples an element $s_0 \in S$ and group elements $g_1, \cdots, g_{C-1} \in_R G$. It outputs a statement $x = (s_0, s_1, \cdots, s_{C-1})$ with $s_i = \alpha(g_i, s_0)$ for $i = 1, \cdots, C-1$, and a witness $w = (g_1, \cdots, g_{C-1})$.
- Algorithm LossyGen randomly samples set elements $s_0, s_1, \cdots, s_{K-1} \in S$ and outputs a lossy statement $x_{\mathsf{ls}} = (s_0, s_1, \cdots, s_{C-1})$.
- On input a statement-witness pair (x, w), $\mathcal{P}_1$ samples a random group element $h \in_R G$ and outputs the commitment $t = \alpha(h, s_0)$.
- On input (x, w, t, c) where $c \in \{0, 1, \cdots, C-1\}$ is a challenge, $\mathcal{P}_2$ outputs a response $f = h * g_c$.
- On input (x, t, c, f), the verification algorithm $\mathcal{V}$ check whether $t = \alpha(f, s_c)$.

Security analysis.

Since the underlying protocol is the same as in Figure 4.1, it is clear that our lossy identification protocol is complete, has α-bit min-entropy with $\alpha \approx \log_2 |O|$, satisfies HVZK property and commitment recoverability. It remains to show that our protocol has indistinguishability of lossy statements, and to calculate the statistical lossy soundness.

Lemma 4.3.3. *Suppose $\alpha : G \times S \to S$ satisfies the C-pseudorandom assumption as in Definition 4.1.2. Then the lossy identification protocol satisfies the lossy statement indistinguishability.*

Proof. The lossy generator of our protocol just random samples C elements $s_0, s_1, \cdots, s_{C-1} \in_R S$. By the hardness assumption of the C-pseudorandom problem, lossy statements and real statements are indistinguishable. □

The following lemma calculates the lossy soundness parameter ϵ_{ls}.

Lemma 4.3.4. *The lossy identification protocol satisfies statistical ϵ_{ls}-lossy soundness for $\epsilon_{\mathsf{ls}} = \frac{1}{C} \prod_{i=1}^{C-1} \frac{A-iB}{A} + \left(1 - \prod_{i=1}^{C-1} \frac{A-iB}{A}\right)$, where $B = |G|$, $A = |S|$.*

Proof. This proof is similar to the proof of [EKP20, Lemma 3.3]. Let $\mathcal{X}$ be the set of the statements such that given a commitment $z \in_R S$, there is only one challenge c resulting in a valid transcript. Consider other commitment z with two valid transcripts (z, c_0, g_0) and (z, c_1, g_1) where these two transcripts satisfy following equations:

$$\alpha(g_0, s_{c_0}) = z$$
$$\alpha(g_1, s_{c_1}) = z.$$

It implies that $\alpha(g_0 * g_1^{-1}, s_{c_0}) = s_{c_1}$, i.e., s_{c_0} and s_{c_1} are in the same orbit. Therefore, if any two elements in the statement are not in the same orbit, the statement can't have two valid transcripts with different challenges.

The number of different statements in $\mathcal{X}$ is $A\prod_{i=1}^{C-1}(A - i|O_i|) \geq A\prod_{i=1}^{C-1}(A - iB)$, where $|O_i|$ is the size of O_i and $|O_i| \leq B$. The number of all statements is A^C. Then we can have the probability that a statement is in $\mathcal{X}$ is $\Pr[x \in \mathcal{X} \mid x \leftarrow \mathsf{LossyGen}] \geq \prod_{i=1}^{C-1}\frac{A-iB}{A}$. We can obtain the probability that an adversary wins as follows:

$$\begin{aligned}
\Pr[\mathcal{A}\text{ wins }] &= \Pr[\mathcal{A}\text{ wins } \mid x \in \mathcal{X}]\Pr[x \in \mathcal{X}] + \Pr[\mathcal{A}\text{ wins } \mid x \notin \mathcal{X}]\Pr[x \notin \mathcal{X}] \\
&\leq \Pr[\mathcal{A}\text{ wins } \mid x \in \mathcal{X}]\Pr[x \in \mathcal{X}] + \Pr[x \notin \mathcal{X}] \\
&= \Pr[\mathcal{A}\text{ wins } \mid x \in \mathcal{X}]\Pr[x \in \mathcal{X}] + (1 - \Pr[x \in \mathcal{X}]) \\
&= (\Pr[\mathcal{A}\text{ wins } \mid x \in \mathcal{X}] - 1)\Pr[x \in \mathcal{X}] + 1 \\
&\leq (\Pr[\mathcal{A}\text{ wins } \mid x \in \mathcal{X}] - 1)\prod_{i=1}^{C-1}\frac{A - iB}{A} + 1 \\
&= \Pr[\mathcal{A}\text{ wins } \mid x \in \mathcal{X}]\prod_{i=1}^{C-1}\frac{A - iB}{A} + \left(1 - \prod_{i=1}^{C-1}\frac{A - iB}{A}\right).
\end{aligned}$$

Note that the second inquality is due to $\Pr[\mathcal{A}\text{ wins } \mid x \in \mathcal{X}] - 1 \leq 0$. This completes the proof. □

Lemma 4.3.4 implies the following for a t-parallel repetition of the lossy identification protocol.

Corollary 4.3.5. *The lossy identification protocol in Figure 4.1, that is run t parallel rounds with the same statement-witness pair, satisfies statistical ϵ_{ls}-lossy soundness for $\epsilon_{\mathsf{ls}} = \frac{1}{C^t}\prod_{i=1}^{C-1}\frac{A-iB}{A} + \left(1 - \prod_{i=1}^{C-1}\frac{A-iB}{A}\right)$, where $A = |S|$, $B = |G|$, and $|C|$ is the size of the challenge space.*

Remark 4.3.6. For our ATFE instantiation in Section 4.4, B is the order of the general linear group $\mathrm{GL}(n, q)$ and A is the size of $\mathrm{ATF}(n, q)$ which is far greater than B as the parameter n is large enough. Therefore, the error ϵ_{ls} is estimated to be $2^{-\lambda}$ where λ is the security level; see Section 4.4 for the detail.

4.3.3 Tightly secure signature scheme in QROM from abstract group actions

A digital signature scheme can be obtained by applying the Fiat-Shamir transformation to the t-fold parallel repetition of the lossy identification protocol in Section 4.3.2. We call this the $\alpha(\mathrm{G}, \mathrm{S})$-GMW-FS-lossy scheme. Note that this result is essentially the same scheme as the $\alpha(\mathrm{G}, \mathrm{S})$-GMW-FS scheme, as the additional LossyGen algorithm used for lossy key generation is only used for security analysis.

We now prove the QROM security of $\alpha(\mathrm{G}, \mathrm{S})$-GMW-FS-lossy based on the C-pseudorandom assumption and the computational unique response assumption as in Lemma 4.2.4.

Theorem 4.3.7. *For any quantum adversary $\mathcal{A}$ against the sEUF-CMA security of $\alpha(\mathrm{G}, \mathrm{S})$-GMW-FS-lossy that issues at most Q_H queries to the quantum random oracle, there exists a quantum adversary $\mathcal{B}$ against the C-pseudorandomness (Definition 4.1.2), a quantum adversary C against the $\alpha(\mathrm{G}, \mathrm{S})$-stabilizer problem (Definition 4.2.3) such that*

$$\begin{aligned} & \mathsf{Adv}_{\mathcal{A}}^{\alpha(\mathrm{G},\mathrm{S})\text{-}GMW\text{-}FS\text{-}lossy\text{-}sEUF\text{-}CMA(\lambda)} \\ \leq\ & \mathsf{Adv}_{\mathcal{B}}^{C\text{-}pseudorandom}(\lambda) + \frac{2}{|O|} \end{aligned}$$

$$+ 8(Q_H+1)^2 \cdot \left(\frac{1}{C^t} \prod_{i=1}^{C-1} \frac{A-iB}{A} + \left(1 - \prod_{i=1}^{C-1} \frac{A-iB}{A} \right) \right)$$
$$+ \mathsf{Adv}_C^{\alpha(\mathrm{G,S})\text{-}Stab}(\lambda)$$

and $\mathsf{Time}(\mathcal{B}) = \mathsf{Time}(\mathcal{A}) + Q_H \cong \mathsf{Time}(\mathcal{A})$. *Here* $B = |G|$, $A = |S|$, *and* $|\mathcal{O}|$ *is the size of the orbit where elements of the statement* $x = (s_0, s_1, \cdots, s_{C-1})$ *are in.*

In the classical setting, we can replace $8(Q_H+1)^2$ *with* $Q_H + 1$.

Proof. The proof initializes with Lemma 4.2.4 and Section 4.3.2 that the underlying sigma protocol has computational unique response, lossiness, lossy-soundness, perfect HVZK and at least $\log(|\mathcal{O}|)$ bits of min-entropy. The result now follows from Theorem 4.3.2. □

4.4 The QROM security of the ALTEQ scheme

Based on the results in Sections 4.2, there are two approaches to show the QROM security of the ALTEQ scheme.

QROM security via perfect unique response.

Let $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ be an alternating trilinear form. Recall that $\mathrm{Stab}(\phi) := \{A \in \mathrm{GL}(n,q) \mid \phi \circ A = \phi\}$.

By Lemma 4.2.1, the ATFE-GMW-FS-$\mathcal{O}(\phi)$ is secure in the quantum model, if $\mathrm{Stab}(\phi)$ is trivial and assume the C-one-way-$\mathcal{O}(\phi)$, where ϕ is instantiated as an alternating trilinear form. To decide whether $\mathrm{Stab}(\phi)$ is trivial or not is a difficult algorithmic problem. Still, we make progress by running experiments for those n of interest in our context.

Basic facts about $\mathrm{Stab}(\phi)$**.** First, note that if $3|q-1$, then $\mathrm{Stab}(\phi)$ cannot be trivial. This is because $3|q-1$ implies the existence of $\lambda \in \mathbb{F}_q$, $\lambda \neq 1$, and $\lambda^3 = 1$. Therefore $\lambda I_n \in \mathrm{Aut}(\phi)$. Second, for (a) $n = 7$ and (b) $n = 8$ and $\mathrm{char}(\mathbb{F}_q) \neq 3$, there exist no alternating trilinear forms with trivial automorphism groups, by classifications of

alternating trilinear forms in these cases [CH88, MN13, HP15]. Third, for $n = 9$ and $q = 2$, by the classification of alternating trilinear forms [HP21], there exists a unique orbit of alternating trilinear forms with trivial automorphism groups.

In general, because of the difference between the dimension of $\mathrm{GL}(n, q)$ (which is n^2) and the dimension of $\mathrm{ATF}(n, q)$ (which is $\binom{n}{3}$), it is expected that for $n \geq 10$ and $3 \nmid q-1$, most alternating trilinear forms would have the trivial automorphism group.

A Magma program to compute the stabilizer group order. We implemented a program in Magma [BJP97] for computing automorphism group orders of alternating trilinear forms as follows.

(1) Enumerate every $v \in \mathbb{F}_q^n$ and compute the rank of $\phi(v, \cdot, \cdot)$ as an alternating bilinear form. Let $S \subseteq \mathbb{F}_q^n$ be the set of non-zero vectors such that $\phi(v, \cdot, \cdot)$ is of lowest rank.

(2) Fix $u \in S$. Let X and Y be two $n \times n$ variable matrices. For every $v \in S$, set up a system of polynomial equations expressing the following:

a) $\phi \circ X = \phi$, and $\phi = \phi \circ Y$.

b) For any $a, b, c \in \mathbb{F}^n$, $\phi(X(a), X(b), c) = \phi(a, b, Y(c))$, and $\phi(X(a), b, c) = \phi(a, Y(b), Y(c))$.

c) $XY = I_n$, and $YX = I_n$.

d) $X(u) = v$, and $Y(v) = u$.

The use the Gröbner basis algorithm implemented in Magma to compute the number of solutions to this system of polynomial equations. Let it be s_v.

(3) Sum over s_v over $v \in S$ as the order of $\mathrm{Stab}(\phi)$.

This algorithm runs in time $q^n \cdot \mathrm{poly}(n, \log q)$. The use of Gröbner computations follows the practices of works in multivariate cryptography for solving polynomial isomorphism [FP06, Bou11, BFFP11, BFV13]. The reason for Step 1 is to limit the number

of Gröbner basis computations, which are more costly compared to computing the ranks. This idea could be found, for example, in [BLQW20].

Report on the results. Our experiment results are as follows.

- For $q = 2$ and $n = 9$, out of 100 samples there are three ones with trivial stabilizer groups. This is consistent with the fact that there exists exactly one orbit of alternating trilinear forms [HP21], so the probability of sampling one from this orbit is $|\mathrm{GL}(2, 9)|/2^{84} \approx 3.6169\%$.
- For $q = 2$ and $n = 10, 11$, all 100 samples return trivial stabilizer groups.
- For $q = 3$ and $n = 10, 11$, all 10 samples return trivial stabilizer groups.

These suggest that for $n = 10$ and q satisfying $3 \nmid q - 1$, a random alternating trilinear form has the trivial automorphism group with good probability. This also implies that for larger n and q such that $3 \nmid q-1$, a random alternating trilinear form has the trivial automorphism group with high probability, as the gap between the space dimension and the group dimension becomes larger as n increases. To the best of our knowledge, to give an estimation of this probability (depending on q and n) is an open problem.

QROM security via lossy schemes. In the above, we presented evidence for the ALTEQ to satisfy the perfect unique response property for $n \geq 10$, supporting its QROM security by the results in Section 4.2. However, the reduction in this approach is not tight. Instead, the QROM security via the lossy scheme approach gives a tight reduction.

To apply the results in Section 4.3 to the ALTEQ scheme, we need to examine whether the group action underlying ATFE is pseudorandom. In Conjecture 3.3.9, we conjectured that this is indeed the case, and provided some supporting evidence, some of which traced back to [JQSY19]. Here we briefly explain that, a key argument is that there seem no easy-to-compute isomorphism invariants for ATFE, as such isomorphism invariants can be used to distinguish non-equivalent alternating trilinear forms.

If the above holds, then $B = |\mathrm{GL}(n,q)| \approx q^{n^2}$ and $A = |\mathrm{ATF}(n,q)| = q^{\binom{n}{3}}$, $A \gg B$ as the security parameter λ is large enough. Therefore, the lossy soundness $\epsilon_{ls} \approx \frac{1}{C^t} \approx \frac{1}{2^\lambda}$.

Lossy schemes with unbalanced challenges. Recall the unbalanced challenge technique described in Section 3.5. The idea is to observe that, in the case of challenge 0, the response would be a random group element that can be expanded from a short seed, so sending the seed reduces the communication. As a result, the number of rounds needs to be increased. This is a standard technique that turns out to be useful in practice as witnessed in [BBPS21, CNP+23b, BDN+23].

Recall the parameters involved in the ALTEQ scheme with unbalanced challenges are as follows. Let M be the round number, K be the number of non-zero challenges, and C the number of alternating trilinear forms in each round. To achieve λ-bit security, we should choose the proper M and K such that $\binom{M}{K} \cdot (C-1)^K \geqslant 2^\lambda$. Some care is then needed to demonstrate the lossy soundness in this setting.

Corollary 4.4.1. *The lossy identification protocol based on ATFE with the unbalanced challenge, satisfies statistical ϵ_{ls}-lossy soundness for*

$$\epsilon_{\mathrm{ls}} = \frac{1}{\binom{M}{K}(C-1)^K} \prod_{i=1}^{C-1} \frac{A - iB}{A} + \left(1 - \prod_{i=1}^{C-1} \frac{A - iB}{A}\right),$$

where $A = |\mathrm{ATF}(n,q)|$, $B = |\mathrm{GL}(n,q)|$.

Proof. Since the size of the challenge space is $\binom{M}{K}(C-1)^K$, we have that $\Pr[\mathcal{A} \text{ wins} \mid x \in \mathcal{X}] \leq \frac{1}{\binom{M}{K}(C-1)^K}$. The result follows the proof for Lemma 4.3.4. □

Chapter 5

Linkable ring signatures based on group action

5.1 Chapter preliminaries

5.1.1 Ring signatures

In this section, we provide the definition of the ring signature.

Definition 5.1.1 (Ring signature)**.** A ring signature scheme Π_{RS} consists of three *PPT* algorithms $(\mathsf{RS.KeyGen}, \mathsf{RS.Sign}, \mathsf{RS.Verify})$ where,

- $\mathsf{RS.SetUp}(1^\lambda)$: Given a security parameter λ, this algorithm outputs the corresponding public parameters pp.
- $\mathsf{RS.KeyGen}(\mathsf{pp})$: This algorithm generates, for a user i, a pair $(\mathsf{vk}_i, \mathsf{sk}_i)$ of the secret key sk_i and public key (verification key) vk_i.
- $\mathsf{RS.Sign}(\mathsf{sk}_i, \mathsf{R}, \mathsf{M})$: Given the secret key sk_i, a list of public keys $\mathsf{R} = \{\mathsf{vk}_1, \ldots, \mathsf{vk}_N\}$ and a message M, it outputs a signature σ.
- $\mathsf{RS.Verify}(\mathsf{R}, \mathsf{M}, \sigma)$: Given a list of public key $\mathsf{R} = \{\mathsf{vk}_1, \ldots, \mathsf{vk}_N\}$, a message M and a signature σ, this algorithm output 1 if this signature is valid or 0 otherwise.

A ring signature needs to satisfy three properties: correctness, anonymity and unforgeability.

Correctness. A ring signature Π_{RS} is said to be correct if for any security parameter λ, polynomial $N = \text{poly}$, any message M, $\mathsf{pp} \leftarrow \mathsf{RS.SetUp}(1^\lambda)$, $(\mathsf{vk}_1, \mathsf{sk}_1), \ldots, (\mathsf{vk}_N, \mathsf{sk}_N) \leftarrow \mathsf{RS.KeyGen}(\mathsf{pp})$, $\sigma \leftarrow \mathsf{RS.Sign}(\mathsf{sk}_i, \mathsf{R}, \mathsf{M})$ with $\mathsf{R} := \{\mathsf{vk}_1, \ldots, \mathsf{vk}_N\}$, it always holds that $\mathsf{RS.Verify}(\mathsf{R}, \mathsf{M}, \sigma) = 1$.

Anonymity. A ring signature Π_{RS} is said to be anonymous if for every security parameter λ and polynomial $N = \text{poly}$, any PPT adversary $\mathcal{A}$ has at most negligible advantage in the following game:

(1) The challenger runs $\mathsf{pp} \leftarrow \mathsf{RS.SetUp}(1^\lambda)$ and generates key pairs $(\mathsf{vk}_i, \mathsf{sk}_i) \leftarrow \mathsf{RS.KeyGen}(\mathsf{pp})$ for all $i \in [N]$ and samples $b \leftarrow_R \{0, 1\}$. Then it sends pp and the secret keys $\{\mathsf{sk}_i\}_{i\in[N]}$ to $\mathcal{A}$.

(2) $\mathcal{A}$ computes a challenge $(\mathsf{R}, \mathsf{M}, i_0, i_1)$, where R contains vk_{i_0} and vk_{i_1}, and sends it to the challenger.

(3) The challenger runs $\mathsf{RS.Sign}(\mathsf{sk}_{i_b}, \mathsf{R}, \mathsf{M}) \rightarrow \sigma$ and sends σ to $\mathcal{A}$.

(4) $\mathcal{A}$ outputs b'. If $b = b'$, then we say that $\mathcal{A}$ wins this game.

The advantage of $\mathcal{A}$ is

$$\mathsf{Adv}_{\mathsf{RS}}^{\mathsf{Anon}}(\mathcal{A}) = |\Pr[\mathcal{A} \text{ wins}] - 1/2| \,.$$

Unforgeability. A ring signature Π_{RS} is said to be unforgeable if for every security parameter λ and polynomial $N = \text{poly}$, any PPT adversary $\mathcal{A}$ has at most negligible probability to win the following game:

(1) The challenger runs $\mathsf{pp} \leftarrow \mathsf{RS.SetUp}(1^\lambda)$ and generates key pairs $(\mathsf{vk}_i, \mathsf{sk}_i) \leftarrow \mathsf{RS.KeyGen}(\mathsf{pp})$ for all $i \in [N]$. It sends the list of public keys $\mathsf{VK} = \{\mathsf{vk}_i\}_{i\in[N]}$ to $\mathcal{A}$ and prepares two empty list SL and CL.

(2) $\mathcal{A}$ can make polynomial times of signing queries and corrupting queries:

- $(\mathsf{sign}, i, \mathsf{R}, \mathsf{M})$: The challenger outputs the signature $\sigma \leftarrow \mathsf{RS.Sign}(\mathsf{sk}_i, \mathsf{R}, \mathsf{M})$ to $\mathcal{A}$ and adds $(i, \mathsf{R}, \mathsf{M})$ to SL.

– (corrupt, i) The challenger sends sk_i to $\mathcal{A}$ and adds vk_i to CL.

(3) We say $\mathcal{A}$ wins this game if $\mathcal{A}$ outputs $(\mathsf{R}', \mathsf{M}', \sigma')$ such that $\mathsf{R}' \subseteq \mathsf{VK} \setminus \mathsf{CL}$, $(\cdot, \mathsf{R}', \mathsf{M}') \notin \mathsf{SL}$, and $\mathsf{RS.Verify}(\mathsf{R}', \mathsf{M}', \sigma') = 1$.

5.1.2 Linkable ring signatures

A linkable ring signature is a variant of a ring signature in which the linkability can detect if a secret key is used more than once. The definition and properties of linkable ring signature, following [BKP20], are provided as follows.

Definition 5.1.2 (Linkable ring signature)**.** A linkable ring signature scheme Π_{LRS} consists of three *PPT* algorithms in the ring signature in addition with a *PPT* algorithm such that:

- $\mathsf{LRS.Link}(\sigma_0, \sigma_1)$: It checks if two signatures σ_0, σ_1 are produced with a same secret key, and outputs 1 if it is the case and 0 otherwise.

Correctness: A linkable ring signature Π_{LRS} is said to have correctness if for any security parameter λ, polynomial $N = \mathrm{poly}$, two messages $\mathsf{M}_0, \mathsf{M}_1$, two sets $D_0, D_1 \subseteq [N]$ such that $j \in D_0 \cap D_1$, $\mathsf{pp} \leftarrow \mathsf{LRS.SetUp}(1^\lambda)$, $\{(\mathsf{vk}_1, \mathsf{sk}_1), \ldots, (\mathsf{vk}_N, \mathsf{sk}_N)\} \leftarrow \mathsf{RS.KeyGen}(\mathsf{pp})$, a random bit $b \leftarrow \{0, 1\}$, $\sigma_b \leftarrow \mathsf{LRS.Sign}(\mathsf{sk}_j, \mathsf{R}_b, \mathsf{M}_b)$ with $\mathsf{R}_b := \{\mathsf{vk}_i\}_{i \in D_b}$, it always holds that $\mathsf{LRS.Verify}(\mathsf{R}, \mathsf{M}, \sigma_b) = 1$ and $\mathsf{LRS.Link}(\sigma_0, \sigma_1) = 1$.

Linkability: A ring signature Π_{LRS} is said to be unforgeable if for every security parameter λ and polynomial $N = \mathrm{poly}$, any PPT adversary $\mathcal{A}$ has at most negligible probability to win the following game:

(1) The challenger runs $\mathsf{pp} \leftarrow \mathsf{LRS.SetUp}(1^\lambda)$ and send pp to $\mathcal{A}$.

(2) $\mathcal{A}$ generates public keys and secret keys $(\{\mathsf{vk}_i, \mathsf{sk}_i\}) \leftarrow \mathsf{LRS.KeyGen}(\mathsf{pp}))$ for $i \in [N]$, and then produces a set $(\sigma_i, \mathsf{M}_i, \mathsf{R}_i)_{i \in [N+1]}$.

(3) We say $\mathcal{A}$ wins this game if all the following conditions are satisfied:

– $\forall i \in [N+1]$, have $\mathsf{R}_i \subseteq \mathsf{VK}$;

– $\forall i \in [N+1]$, have $\mathsf{LRS.Verify}(\mathsf{R}_i, \mathsf{M}_i, \sigma_i) = 1$;

- $\forall i, j \in [N+1]$, where $i \neq j$, have $\mathsf{LRS.Link}(\sigma_i, \sigma_j) = 0$.

Linkable Anonymity: A ring signature Π_{LRS} is said to be linkable anonymous if for every security parameter λ and polynomial $N = \text{poly}$, any PPT adversary $\mathcal{A}$ has at most negligible advantage in the following game:

(1) The challenger runs $\mathsf{pp} \leftarrow \mathsf{LRS.SetUp}(1^\lambda)$ generates public keys and secret keys $(\{\mathsf{vk}_i, \mathsf{sk}_i\}) \leftarrow \mathsf{RS.KeyGen}(\mathsf{pp}; \mathsf{rr}_i)$ for $i \in [N]$ using random coins rr_i and it also samples a ramdom bit $b \in \{0, 1\}$. Then it sends the public keys $\mathsf{VK} = \{\mathsf{vk}_0, \ldots, \mathsf{vk}_N\}$ to $\mathcal{A}$.

(2) $\mathcal{A}$ sends two public keys $\mathsf{vk}'_0, \mathsf{vk}'_1$ to the challenger, and we let $\mathsf{sk}'_0, \mathsf{sk}'_1$ be the corresponding secret keys.

(3) The challenger outputs rr_i of the corresponding $\mathsf{vk}_i \subseteq \mathsf{VK} \setminus \{\mathsf{vk}'_0, \mathsf{vk}'_1\}$.

(4) $\mathcal{A}$ chooses a public key $\mathsf{vk} \in \{\mathsf{vk}'_0, \mathsf{vk}'_1\}$ and provides a message M and a ring R that $\{\mathsf{vk}'_0, \mathsf{vk}'_1\} \subseteq \mathsf{R}$ to query the challenger:
   - If $\mathsf{vk} = \mathsf{vk}'_0$, the challenger outputs the signature $\mathsf{LRS.Sign}(sk_b, \mathsf{R}, \mathsf{M}) \rightarrow \sigma$.
   - If $\mathsf{vk} = \mathsf{vk}'_1$, the challenger outputs the signature $\mathsf{LRS.Sign}(sk_{1-b}, \mathsf{R}, \mathsf{M}) \rightarrow \sigma$.

(5) $\mathcal{A}$ output a guess b'. If $b = b'$, we say $\mathcal{A}$ wins this game.

The advantage of $\mathcal{A}$ is $\mathsf{Adv}^{\mathsf{Anon}}_{\mathsf{LRS}}(\mathcal{A}) = |\Pr[\mathcal{A} \text{ wins}] - 1/2|$.

Non-frameability: A ring signature Π_{LRS} is said to be non-frameable if for every security parameter λ and polynomial $N = \text{poly}$, any PPT adversary $\mathcal{A}$ has at most negligible probability to win the following game:

(1) The challenger runs $\mathsf{pp} \leftarrow \mathsf{LRS.SetUp}(1^\lambda)$ generates public keys and secret keys $(\{\mathsf{vk}_i, \mathsf{sk}_i\}) \leftarrow \mathsf{RS.KeyGen}(\mathsf{pp})$ for $i \in [N]$. It sends the list of public keys $\mathsf{VK} = \{\mathsf{vk}_i\}_{i \in [N]}$ to $\mathcal{A}$ and prepares two empty list SL and CL.

(2) $\mathcal{A}$ can make polynomial times of signing queries and corrupting queries:
   - $(\mathsf{sign}, i, \mathsf{R}, \mathsf{M})$: The challenger outputs the signature $\mathsf{LRS.Sign}(\mathsf{sk}_i, \mathsf{R}, \mathsf{M}) \rightarrow \sigma$ to $\mathcal{A}$ and adds $(i, \mathsf{R}, \mathsf{M})$ to SL.

- (corrupt, i): The challenger sends the random bits r_i to $\mathcal{A}$ and adds vk_i to CL.

(3) We say $\mathcal{A}$ wins this game if $\mathcal{A}$ outputs $(\mathsf{R}', \mathsf{M}', \sigma')$ such that $(\cdot, \mathsf{M}', \mathsf{R}') \notin \mathsf{SL}$, $\mathsf{LRS.Verify}(\mathsf{R}', \mathsf{M}', \sigma') = 1$, and for some query $(i, \mathsf{R}, \mathsf{M}) \in \mathsf{SL}$ where the identity i satisfies $\mathsf{vk}_i \in \mathsf{VK} \setminus \mathsf{CL}$, the challenger outputs a signature σ that $\mathsf{LRS.Link}(\sigma', \sigma) = 1$ holds.

Unforgeability: The definition of unforgeability remains the same as that of the normal ring signature. The unforgeability can be easily derived from the linkable anonymity and the non-frameability.

5.2 Ring signatures from abstract group actions

In this section, we describe the construction of linkable ring signatures from abstract group actions. It follows the framework of Beullens, Katsumata and Pintore [BKP20], so we call it the GMW-FS-BKP design. While [BKP20] focussed on commutative group actions, their ring signature construction readily applies to general group actions. In fact, for our group action framework, the scheme becomes a bit simpler because [BKP20] needs to work with rejection sampling.

5.2.1 Base OR-Sigma protocol from abstract group actions

Before introducing the base OR-Sigma protocol, let's recall a useful tree structure, namely the index-hiding Merkle tree from [BKP20].

Index-hiding Merkle trees. Merkle trees [Mer89] are used widely in signature scheme designs, including recent works on isogeny-based signatures [BKV19, FG19]. It provides a method for hashing a list of elements $A = (a_0, \ldots, a_N)$ into a single hash value, commonly called the root. Subsequently, one can efficiently demonstrate to a third party that a specific element a_i is part of the list A at a certain position. In the following discussion, we consider a slightly modified version of the standard Merkle

$\mathcal{P}_1(s_1, \ldots, s_N)$

1 : seed $\leftarrow_R \{0,1\}^\lambda$
2 : $(h, \text{bits}_1, \ldots, \text{bits}_N) \leftarrow \text{PRG}(\text{seed})$
3 : **for** i from 1 to N **do**
4 : $\quad t_i \leftarrow \alpha(h, s_i)$
5 : $\quad \text{C}_i \leftarrow \text{Com}(t_i, \text{bits}_i)$
6 : $(\text{root}, \text{tree}) \leftarrow \text{MerkleTree}(\text{C}_1, \ldots, \text{C}_N)$
7 : com $\leftarrow$ root
8 : The prover $\mathcal{P}$ sends the commitment com to the verifier $\mathcal{V}$

$\mathcal{V}_1(\text{com})$

1 : $c \leftarrow_R \{0,1\}$
2 : cha $\leftarrow c$
3 : The verifier $\mathcal{V}$ sends the challenge cha to the prover $\mathcal{P}$

$\mathcal{P}_2(g_I, I, \text{cha})$

1 : $c \leftarrow$ cha
2 : **if** $c = 0$ **then**
3 : $\quad f \leftarrow h * g_I$
4 : $\quad \text{path} \leftarrow \text{getMerklePath}(\text{tree}, I)$
5 : $\quad \text{rsp} \leftarrow (f, \text{path}, \text{bits}_I)$
6 : **else**
7 : $\quad \text{rsp} \leftarrow \text{seed}$
8 : The prover $\mathcal{P}$ sends the response rsp to the verifier $\mathcal{V}$

$\mathcal{V}_2(\text{com}, \text{cha}, \text{rsp}, s_0, s_1, \ldots, s_N)$

1 : $(\text{root}, c) \leftarrow (\text{com}, \text{cha})$
2 : **if** $c = 0$ **then**
3 : $\quad (f, \text{path}, \text{bits}) \leftarrow \text{rsp}$
4 : $\quad \tilde{t} \leftarrow \alpha(f, s_0)$
5 : $\quad \tilde{\text{C}} \leftarrow \text{Com}(\tilde{t}, \text{bits})$
6 : $\quad \widetilde{\text{root}} \leftarrow \text{ReconstructRoot}(\tilde{\text{C}}, \text{path})$
7 : $\quad$ The verifier $\mathcal{V}$ outputs accept if $\widetilde{\text{root}} = \text{root}$, else outputs reject
8 : **else**
9 : $\quad \text{seed} \leftarrow \text{rsp}$
10 : $\quad \widetilde{\text{root}} \leftarrow \mathcal{P}_1((s_1, \ldots, s_N), \text{seed})$
11 : $\quad$ The verifier $\mathcal{V}$ outputs accept if $\widetilde{\text{root}} = \text{root}$, else outputs reject

Figure 5.1: OR-Sigma protocol.

tree. This modification enables the proof of inclusion of a specific element a_i without disclosing its exact position within the list. Formally, the Merkle tree consists of three algorithms (MerkleTree, getMerklePath, ReconstructRoot) with a hash function $\mathcal{H}_{\mathsf{Coll}} : \{0,1\}^* \to \{0,1\}^{2\lambda}$.

- $\mathsf{MerkleTree}(A) \to (\mathsf{root}, \mathsf{tree})$: On input a list of 2^k elements $A = (a_1, \ldots, a_{2^k})$ with $k \in \mathbb{N}$, it constructs a binary tree of height k. Its leaves consist of the hash values of the list elements, and each internal node is equal to the concatenated hash value of its two children, e.g. $\mathcal{H}_{\mathsf{Coll}}(h_{\mathsf{left}}||h_{\mathsf{right}})$. Here we consider another way of concatenation, that is, sorting the two child nodes according to the lexicographical order, e.g. $\mathcal{H}_{\mathsf{Coll}}((h_{\mathsf{left}}||h_{\mathsf{right}})_{\mathsf{lex}})$. The algorithm then outputs the root root of the Merkle tree, as well as a description of the entire tree tree.
- $\mathsf{getMerklePath}(\mathsf{tree}, I) \to \mathsf{path}$: On input the description of a Merkle tree tree and an index $i \in [2^k]$, it outputs the list path, which contains k nodes in the tree. They are sorted by the siblings of $\mathcal{H}_{\mathsf{Coll}}(a_I)$ and their parents' siblings.
- $\mathsf{ReconstructRoot}(a, \mathsf{path}) \to \mathsf{root}$: On input an element a in the list of elements $A = (a_1, \ldots, a_{2^k})$ and $\mathsf{path} = (n_1, \ldots, n_k)$, it outputs a reconstructed root (a hash value), which is calculated by putting $h_0 = \mathcal{H}_{\mathsf{Coll}}(a)$ and defining h_i for $i \in [k]$ recursively as $h_i = \mathcal{H}_{\mathsf{Coll}}((h_{i-1}, n_i)_{\mathsf{lex}})$.

The Beullens-Katsumata-Pintore design. Briefly speaking, the GMW-FS-BKP ring signature is obtained by applying the Fiat-Shamir transformation to an OR-Sigma protocol, which is an interactive protocol in which a prover convinces a verifier that she knows the witness of one of several given inputs without revealing which one. Here, we describe the base OR-Sigma protocol (see Figure 5.1) for an abstract group action.

Let $g_1, g_2, \ldots, g_N \leftarrow_R G$ be the secret keys, and $s_1 = \alpha(g_1, s_0), \ldots, s_N = \alpha(g_N, s_0)$ be the public keys, Com be a commitment scheme. The base OR-Sigma protocol with *statement* $\{s_0, \ldots, s_N \in S\}$ and *witness* $\{g_I \in G, I \in [N]$ such that $\alpha(g_I, s_0) = s_I\}$, works as follows.

(1) First, the prover random sample a group element $h \in G$, and apply it to $s_1, \ldots, s_N$ respectively. Specifically, $t_1 = \alpha(h, s_1), \ldots, t_N = \alpha(h, s_N)$. Then the prover samples $\text{bits}_i \leftarrow_R \{0,1\}^\lambda$ and commits to t_i with $\mathsf{C}_i = \mathsf{Com}(t_i, \text{bits}_i)$. The prover further builds a Merkle tree with the $(\mathsf{C}_1, \ldots, \mathsf{C}_N)$ as its leaves. The prover computes the root root of the Merkle tree and sends it to the verifier as the commitment.

(2) When the verifier receives the commitment, it will randomly sample a challenge $c \leftarrow_R \{0,1\}$ and respond to the prover.

(3) If $c = 0$, then the prover computes $f = h * g_I$ and the authenticated path for C_I. The prover sends back a response $\text{rsp} = (f, \text{path}, \text{bits}_I)$. The verifier applies f to s_0 to get $\tilde{t}$ and computes $\tilde{\mathsf{C}} = \mathsf{Com}(\tilde{t}, \text{bits}_I)$. The verifier then get a root $\widetilde{\text{root}}$ by path and $\tilde{\mathsf{C}}$. Finally the verifier checks whether $\widetilde{\text{root}} = \text{root}$.

(4) If $c = 1$, then the prover sends $(h, \text{bits}_1, \ldots, \text{bits}_N)$ to the verifier. This information allows the verifier to rebuild a Merkle tree as in step 1, and then check that the roots are consistent.

5.2.2 Optimization

Following some optimization techniques used in [BKP20], we can have a more efficient OR-Sigma protocol. We just briefly describe the following three techniques, for more details please see [BKP20, Section 3.4].

(1) The challenge space of the original challenge space is binary. One can observe that the response with challenge $\text{cha} = 0$ is more costly than that challenge $\text{cha} = 1$. Instead of choosing the challenge bit uniformly in each round, we execute OR sigma protocol $M > \lambda$ rounds and fix exactly K rounds with challenge $\text{cha} = 0$. To satisfy the λ bits of security, we can choose proper parameters M, K such that $\binom{M}{K} \geq 2^\lambda$. Denote $C_{M,K}$ as the set of strings in $\{0,1\}^M$ with K-bits of 0.

(2) With the unbalanced challenge space technique, we do M executions of OR sigma protocol and $M - K$ executions respond with random seeds. Instead of randomly sampling M independent seeds, we can utilize a seed tree to generate these seeds. Then prover can responsd with $\mathsf{seeds}_{\mathsf{internal}} \leftarrow \mathsf{ReleaseSeeds}(\mathsf{seed}_{\mathsf{root}}, c)$ instead of $M - K$ seeds, where $\mathbf{c}$ is randomly sampled from $C_{M,K}$. The verifier can use $\mathsf{seeds}_{\mathsf{internal}}$ and $\mathbf{c}$ to recover $M - K$ seeds by $\mathsf{RecoverLeaves}(\mathsf{seeds}_{\mathsf{internal}}, \mathbf{c})$. Note that here we divide M leaves into K parts, and put a leaf corresponding to $\mathbf{c}_{i,i\in[M]} = 0$ in each part, which leads to a smaller upper bound $K \cdot \log_2(\frac{M}{K})$ for the internal seeds.

(3) Adding salt is a well-known technique that allows us to have tighter security proofs for zero-knowledge. Also, it avoids multi-target attacks, as in [DN19], without affecting too much efficiency.

After applying the above methods, we obtain the optimized base OR sigma protocol shown in Figure 5.2 where we simplify internal seeds $\mathsf{seeds}_{\mathsf{internal}}$ as $\mathsf{seeds}_{\mathsf{int}}$, the SeedTree function as Sd, the ReleaseSeeds function as Rls, the RecoverLeaves function as Rcv, the seed expander and the commitment scheme $\mathcal{O}(\mathsf{salt}||\cdot)$ with salt as $\mathcal{O}_{\mathsf{s}}$ and the seed expander and the commitment scheme $\mathcal{O}(\mathsf{salt}||i||\cdot)$ with salt and the ith instance as $\mathcal{O}_{si}$.

Note that the group action α with one-way assumption satisfies the definition of *admissible group action* in [BKP20]. Then we prove the security of the optimized base OR-Sigma protocol shown in Figure 5.2 as follows.

Theorem 5.2.1. *Define the following relation*

$$R = \left\{ ((s_0, s_1, \ldots, s_N), (g, I)) \middle| \begin{array}{c} g \in G, s_i \in S \\ I \in [N], s_I = \alpha(g, s_0) \end{array} \right\},$$

$\mathcal{P}_1'(s_1, \ldots, s_N)$

1 : $\text{seed}_{\text{root}} \leftarrow_R \{0,1\}^\lambda$
2 : $\text{salt} \leftarrow_R \{0,1\}^{2\lambda}$
3 : $(\text{seed}_1, \ldots, \text{seed}_M) \leftarrow \text{Sd}^{O_s}(\text{seed}_{\text{root}}, M)$
4 : **for** i from 1 to M **do**
5 : $\quad \text{com}_i \leftarrow \mathcal{P}_1^{O_{si}}((s_1, \ldots, s_N), \text{seed}_i)$
6 : $\text{com} \leftarrow (\text{salt}, \text{com}_1, \ldots, \text{com}_M)$
7 : $\mathcal{P}$ sends com to $\mathcal{V}$

$\mathcal{V}_1'(\text{com})$

1 : $\mathbf{c} \leftarrow_R C_{M,K}$
2 : $\text{cha} \leftarrow \mathbf{c}$
3 : $\mathcal{V}$ sends cha $\mathcal{P}$

$\mathcal{P}_2'(g_I, I, \text{cha})$

1 : $\mathbf{c} = (c_1, \ldots, c_M) \leftarrow \text{cha}$
2 : **for** i s.t. $c_i = 0$ **do**
3 : $\quad \text{rsp}_i \leftarrow \mathcal{P}_2(g_I, I, c_i, \text{seed}_i)$
4 : $\text{seeds}_{\text{int}} \leftarrow \text{Rls}^{O_s}(\text{seed}_{\text{root}}, \mathbf{c})$
5 : $\text{rsp} \leftarrow (\text{seeds}_{\text{int}}, \{\text{rsp}_i\}_{i \text{ s.t. } c_i=0})$
6 : $\mathcal{P}$ sends rsp to $\mathcal{V}$

$\mathcal{V}_2'(\text{com}, \text{cha}, \text{rsp}, s_0, s_1, \ldots, s_N)$

1 : $(\text{salt}, \text{com}_1, \ldots, \text{com}_M) \leftarrow \text{com}$
2 : $\mathbf{c} = (c_1, \ldots, c_M) \leftarrow \text{cha}$
3 : $(\text{seeds}_{\text{int}}, \{\text{rsp}_i\}_{i \text{ s.t. } c_i=0}) \leftarrow \text{rsp}$
4 : $\{\text{rsp}_i\}_{i \text{ s.t. } c_i=1} \leftarrow \text{Rcv}^{O_s}(\text{seeds}_{\text{int}}, \mathbf{c})$
5 : **for** i from 1 to M **do**
6 : $\quad$ **if** $\mathcal{V}_2^{O_{si}}(\text{com}_i, c_i, \text{rsp}_i) = \text{reject}$ **then**
7 : $\quad\quad \mathcal{V}$ outputs reject
8 : $\mathcal{V}$ outputs accept

Figure 5.2: Optimized OR sigma protocol.

and the relaxed relation

$$R = \left\{ ((s_0, s_1, \ldots, s_N), w) \;\middle|\; \begin{array}{rc} & g \in G, s_i \in S \\ w = (g, I): & I \in [N], s_I = \alpha(g, s_0) \\ w = (x, x'): & or\ x \neq x', \mathcal{H}_{\text{Coll}}(x) = \mathcal{H}_{\text{Coll}}(x') \\ & or\ \text{Com}(x) = \text{Com}(x') \end{array} \right\},$$

Then the optimized base OR sigma protocol shown in Figure 5.2 has correctness, relaxed special soundness and honest-verifier zero-knowledge for the relation R.

Proof. Based on the group action one-way assumption, it's straightforward to see that our optimized base OR sigma protocol satisfies the properties in [BKP20, Definition 3.1]. By Theorem 3.5 and Theorem 3.6 in [BKP20], the optimized base OR sigma protocol satisfies correctness, relaxed special soundness and honest-verifier zero-knowledge.

Note that the only difference between our protocol and [BKP20] is that our protocol does not have abort, so the proof is basically the same. We provide a brief proof of the base OR-sigma protocol as follows.

Correctness. If the protocol is executed honestly with the input (g_I, I) such that $\alpha(g_I, s_0) = s_I$, then the verifier accepts with probability 1. We will discuss the two cases of $c = 0$ and $c = 1$ separately as follows. If $c = 0$, we have $\alpha(f, s_0) = \alpha(h * g_I, s_0) = \alpha(h, s_I) = t_I$. Then we can reconstruct the correct root by $\mathsf{Com}(t_I, \mathsf{bits}_I)$. If $c = 1$, the verifier just needs to rebuild a Merkle tree and then obtain a correct root.

Relaxed special soundness. Here we model the commitment scheme Com as a random oracle O'. Then there is an extractor, given two valid transcripts $(\mathsf{root}, 0, (f, \mathsf{path}, \mathsf{bits}))$ and $(\mathsf{root}, 1, \mathsf{seed})$ outputs a witness. There are three possibilities for witness as follows: (1) g such that $\alpha(g, s_0) = s_I$ for some $I \in [N]$; (2) a collision in $\mathcal{H}_{\mathsf{Coll}}$; (3) a collision in O'. The extractor first expands the seed seed to obtain $(h, \mathsf{bits}_1, \ldots, \mathsf{bits}_N)$ and then repeats what $\mathcal{P}_1$ did at the first round. That is, the extractor can rebuild a Merkle tree to obtain root′ from $\mathsf{MerkleTree}(\mathsf{C}_1, \ldots, \mathsf{C}_N)$, where $\mathsf{C}_i = \mathsf{Com}(\alpha(h, s_i), \mathsf{bits}_i)$. For another valid transcript, the extractor use the function $\mathsf{ReconstructRoot}(\tilde{\mathsf{C}}, \mathsf{path})$ to obtain the root root, where $\tilde{\mathsf{C}} = \mathsf{Com}(\alpha(f, s_0), \mathsf{bits})$. We have root = root′ due to two valid transcripts. The extractor then checks whether $\tilde{\mathsf{C}} \neq \mathsf{C}_i$ for all $i \in [N]$, if so $\tilde{\mathsf{C}}$ is not a leaf of this tree, which implies the extractor can output a collision for the hash function $\mathcal{H}_{\mathsf{Coll}}$. Otherwise, if $\tilde{\mathsf{C}} = \mathsf{C}_I$ for some $I \in [N]$, the extractor then checks whether $(\alpha(f, s_0), \mathsf{bits}) \neq (\alpha(h, s_i), \mathsf{bits}_I)$. if so, this implies the extractor can output a collision for the oracle O'. Otherwise, the extractor outputs $\mathsf{wit} = h^{-1} \cdot f$ as the witness, where $\alpha(\mathsf{wit}, s_0) = s_I$.

Honest-verifier zero knowledge. Here, we only give a proof sketch; the complete proof is consistent with [BKP20]. Since the prover does use the witness (the response does not involve the witness) when challenge 1, it is straightforward that the simulator simulates transcripts with challenge 1. When the challenge is 0, the advantage of the adversary over the simulator is negligible, e.g., bounded by $2Q/2^{\lambda}$, where we

assume the adversary who makes Q queries to the random oracle. Here we briefly describe where the $2Q$ factor comes from. We can consider a series of simulators as a proof strategy. Two of the simulators introduce a Q factor each. The first simulator is equal to the honest prover, expect that by sampling $(h, \text{bits}_1, \ldots, \text{bits}_N)$ uniformly at random. This does not change the view of the adversary unless the adversary queried the oracle related to $\text{PRG}(\text{seed})$. Since seed has λ bits of min-entropy and it's information-theoretically hidden from the adversary, then the probability that this happens is bounded by $Q/2^\lambda$. The second simulator is equal to the first simulator except that the commitments C_i for $i \neq I$ are chosen uniformly at random. This also does not change the view of the adversary, unless the adversary queried the oracle related with $\text{Com}(t_i, \text{bits}_i)$ for an $i \in [N]$ with $i \neq I$. The probability that this happens is also bounded by $Q/2^\lambda$. □

5.2.3 From OR-Sigma protocol to ring signatures

In this section, we obtain a ring signature by applying the Fiat-Shamir transformation to the OR-Sigma protocol. The key generation, signature generation and verification of the ring signature scheme are described in Algorithms 4, 5, 6, and 7 respectively.

Algorithm 4: Set Up

Input: The security parameter λ.

Output: Public parameter: an element $s_0 \in S$.

1 Randomly sample an element s_0 from S.

2 **return** *Public parameter:* s_0.

Algorithm 5: Key generation

Input: public parameter s_0, the user i.

Output: Public key for the user i: an element $s_i \in S$.

Private key for the user i: A group element g_i such that $s_i = \alpha(g_i, s_0)$.

1 Randomly sample a group element g_i from G.

2 Compute $s_i \leftarrow \alpha(g_i, s_0)$.

3 **return** *Public key:* s_i. *Private key:* g_i.

Algorithm 6: Signing procedure

Input: The public key $s_0, \ldots, s_N$, the private key g_I of a user $I \in [N]$, a message msg, a commitment scheme $\mathsf{Com} : \{0,1\}^* \to \{0,1\}^\lambda$, a hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\lambda$.

Output: A signature Sig on msg.

1 $\mathsf{com} = (\mathsf{salt}, (\mathsf{com}_i)_{i\in[M]}) \leftarrow \mathcal{P}'_1(s_1, \ldots, s_N)$

2 $\mathsf{cha} \leftarrow \mathcal{H}(\mathsf{msg}||s_1|| \cdots ||s_N||\mathsf{com})$

3 $\mathsf{rsp} \leftarrow \mathcal{P}'_2(g_I, I, \mathsf{cha})$

4 **return** $\mathsf{Sig} = (\mathsf{salt}, \mathsf{cha}, \mathsf{rsp})$

Algorithm 7: Verification procedure

Input: The public key $s_0, \ldots, s_N \in S$. The signature $\mathsf{Sig} = (\mathsf{salt}, \mathsf{cha}, \mathsf{rsp})$. The message msg. A hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\lambda$.

Output: "Yes" if Sig is a valid signature for msg. "No" otherwise.

1 $\mathsf{com} \leftarrow \mathsf{RecoverCom}(s_0, \ldots, s_N, \mathsf{salt}, \mathsf{cha}, \mathsf{rsp})$

2 **if** $\mathsf{accept} = \mathcal{V}'_2(\mathsf{com}, \mathsf{cha}, \mathsf{rsp}) \wedge \mathsf{cha} = \mathcal{H}(\mathsf{msg}||s|| \cdots ||s_N||\mathsf{com})$ **then**

3 **return** *Yes*

4 **else**

5 **return** *No*

Remark 5.2.2. Since the optimized base OR sigma protocol is proved to satisfy all properties in Theorem 5.2.1, the correctness, anonymity and unforgeability of the ring signature are straightforward.

5.3 Linkable ring signatures from abstract group actions

The linkable property. Linkable ring signatures were first introduced by Liu and Wong [LW05] that allow public checking whether two ring signatures are 'linked', i.e., generated by one user. A typical approach to construct a linkable ring signature is to add a tag, which uniquely defines the real signer, to a signature. The approach in [BKP20] is to first construct a linkable OR sigma protocol and then apply Fiat-Shamir transformation to obtain a linkable ring signature.

For this, we add a tag $r_0 \in S$ associated with a group action $\beta : G \times S \longrightarrow S$ into the relation. The group action β is defined as $\beta(g, s) = \alpha(g^{-t}, s)$ where t is an involution of G. This tag r_0 is used to track if some secret key is signed more than once. In addition, we restrict the initial public key s_0 is sampled from an orbit $\mathcal{O}(s_0)$ with a trivial automorphism group. After adding the tag into the base OR sigma protocol, we can get a linkable OR sigma protocol and apply certain optimization methods to it for more efficiency.

To derive the security proof for linkable OR sigma protocol, the following properties of the pair of group actions are needed; see [BKP20, Definition 4.2], and also [BBN+22, CNP+23b].

Definition 5.3.1. Given two group actions $\alpha : G \times S \longrightarrow S$ and $\beta : G \times S \longrightarrow S$. We define the following properties:

(1) Efficiency: One can efficiently compute $\alpha(g, s)$ and $\beta(g, s)$ for any $g \in G$ and $s \in S$, uniformly sample from G and S, and represent elements in G and S uniquely.

(2) Linkability: Given $(s_0, r_0) \in S \times S$, it's hard to produce $g, g' \in G$ such that $\alpha(g, s_0) = \alpha(g', s_0)$ and $\beta(g, r_0) \neq \beta(g', r_0)$

(3) Linkable Anonymity: Given $(s_0, r_0) \in S \times S$, the pair $(s_1, r_1) = (\alpha(g, s_0), \beta(g, r_0))$ is computationally indistinguishable from (s_2, r_2) where $g \in_R G$ and $s_2, r_2 \in_R S$.

(4) Non-Frameability: Given $(s_0, r_0) \in S \times S$, $s_1 = \alpha(g, s_0)$ and $r_1 = \alpha(g, r_0)$, it's hard to find a group element g' such that $r_1 = \alpha(g', r_0)$

The linkable anonymity is captured by the following property about group action pairs.

Definition 5.3.2. Let $\alpha, \beta : G \times S \to S$ be two group actions. We say that the (α, β) pair satisfies the *pseudorandom* assumption at $(s_0, r_0) \in S \times S$, if no probabilistic or quantum polynomial-time algorithms can distinguish the following two distributions with non-negligible probability:

(1) The random distribution: $(s_1, r_1) \in S \times S$, where $s_1, r_1 \leftarrow_R S$.

(2) The pseudorandom distribution: $(s_1, r_1) \in S \times S$, where $g \leftarrow_R G$, and $s_1 = \alpha(g, s_0)$ and $r_1 = \beta(g, r_0)$.

Furthermore, if the group actions α and β also satisfy the trivial stabilizer assumption (Definition 4.2.6), then the linkability and non-frameability also follow. These together suffice to prove the security of the linkable GMW-FS-BKP design based on the action pair (α, β). We note that the above strategy was already used in MEDS [CNP+23b] for the action underlying the matrix code equivalence problem.

Instantiations of pseudorandom group action pairs. Let $\alpha : G \times S \to S$ be a group action. There are some generic recipes in the literature about finding another action $\beta : G \times S \to S$ so that (α, β) is pseudorandom. In [BKP20], β is constructed as $\beta(g, s) = \alpha(g^2, s)$. In [BMPS20, CNP+23b], β is constructed as $\beta(g, s) = \alpha(g^{-1}, s)$. Note that here β is actually a right action (if α is a left action). It follows that the responses need to involve both gh and hg where h is a random group element and g is the secret.

We note that it is possible to do slightly better than the above, if we have an involution t of G, i.e. an anti-automorphism of order 2. This means that t is an automorphism, $(g^t)^t = g$, and $(g * h)^t = h^t * g^t$. We can then define $\beta(g, s) = \alpha(g^{-t}, s)$. In the case of $G = \mathrm{GL}(n, q)$ as of interest in ATFE (and MEDS), this t can be simply taken as

the transpose of matrices. This gives a concrete linkable ring signature scheme based on ATFE-GMW-FS-BKP. Of course, further research is required to verify whether this instantiation does give a pseudorandom group action pair.

$\mathcal{P}_1(s_1, \ldots, s_N, r)$

1 : seed $\leftarrow_R \{0,1\}^\lambda$
2 : $(h, \text{bits}_1, \ldots, \text{bits}_N) \leftarrow \text{PRG}(\text{seed})$
3 : $r' \leftarrow \beta(h, r)$
4 : **for** i from 1 to N **do**
5 : $\quad t_i \leftarrow \alpha(h, s_i)$
6 : $\quad \text{C}_i \leftarrow \text{Com}(t_i, \text{bits}_i)$
7 : $(\text{root}, \text{tree}) \leftarrow \text{MerkleTree}(\text{C}_1, \ldots, \text{C}_N)$
8 : $h \leftarrow \mathcal{H}_{\text{Coll}}(r', \text{root})$
9 : com $\leftarrow h$
10 : $\mathcal{P}$ sends com to $\mathcal{V}$

$\mathcal{V}_1(\text{com})$

1 : $c \leftarrow_R \{0,1\}$
2 : cha $\leftarrow c$
3 : $\mathcal{V}$ sends cha to $\mathcal{P}$

$\mathcal{P}_2(A_I, I, \text{cha})$

1 : $c \leftarrow$ cha
2 : **if** $c = 0$ **then**
3 : $\quad f \leftarrow h * g_I$
4 : $\quad \text{path} \leftarrow \text{getMerklePath}(\text{tree}, I)$
5 : $\quad \text{rsp} \leftarrow (f, \text{path}, \text{bits}_I)$
6 : **else**
7 : $\quad \text{rsp} \leftarrow \text{seed}$
8 : $\mathcal{P}$ sends rsp to $\mathcal{V}$

$\mathcal{V}_2(\text{com}, \text{cha}, \text{rsp}, s_0, s_1, \ldots, s_N, r_0, r)$

1 : $(h, c) \leftarrow (\text{com}, \text{cha})$
2 : **if** $c = 0$ **then**
3 : $\quad (f, \text{path}, \text{bits}) \leftarrow \text{rsp}$
4 : $\quad \tilde{t} \leftarrow \alpha(f, s_0)$
5 : $\quad \tilde{\text{C}} \leftarrow \text{Com}(\tilde{t}, \text{bits})$
6 : $\tilde{r'} \leftarrow \beta(f, r_0)$
7 : $\quad \widetilde{\text{root}} \leftarrow \text{ReconstructRoot}(\tilde{\text{C}}, \text{path})$
8 : $\quad$ **if** $h = \mathcal{H}_{\text{Coll}}(\tilde{r'}, \widetilde{\text{root}})$ **then**
9 : $\quad\quad \mathcal{V}$ outputs accept
10 : $\quad$ **else**
11 : $\quad\quad \mathcal{V}$ outputs reject
12 : **else**
13 : $\quad \text{seed} \leftarrow \text{rsp}$
14 : $\quad \widetilde{\text{root}} \leftarrow \mathcal{P}_1((s_1, \ldots, s_N), \text{seed})$
15 : $\quad$ **if** $h = \mathcal{H}_{\text{Coll}}(\tilde{r'}, \widetilde{\text{root}})$ **then**
16 : $\quad\quad \mathcal{V}$ outputs accept
17 : $\quad$ **else**
18 : $\quad\quad \mathcal{V}$ outputs reject

Figure 5.3: Linkable OR sigma protocol.

Theorem 5.3.3. *Assume the stabilizers* $\text{Stab}(s_0)$ *and* $\text{Stab}(r_0)$ *are trivial (see Definition 4.2.6) and the pair-pseudorandom problem defined in Definition 5.3.2 is hard. The*

linkable OR sigma protocol shown in Figure 5.3 after the optimization satisfies the properties defined in Definition 5.3.1.

Proof. For the linkability, we derive this property by restricting the orbit $O(s_0)$ has a trivial stabilizer. Then by the pair-pseudorandom assumption, it's straightforward to see that our protocol has linkable anonymity. For the non-frameability, we restrict the stabilizer $\mathrm{Stab}(r_0)$ to be trivial as well, and then the group element g satisfying $s_1 = \alpha(g, s_0)$ and $r_1 = \alpha(g, r_0)$ is unique. It follows that if one can break non-frameability, the pair-pseudorandom assumption can be broken as well. □

Corollary 5.3.4. *The linkable OR sigma protocol shown in Figure 5.3 after the optimization satisfies correctness, high min-entropy, special zero-knowledge and relaxed special soundness.*

Proof. By Theorem 5.3.3, our OR sigma protocol satisfies correctness, relaxed special soundness and honest-verifier zero-knowledge. The proof is similar to that in Theorem 5.2.1. Thus we omit the proof here. □

After applying the Fiat-Shamir transformation to the linkable OR sigma protocol, we obtain a linkable ring signature shown in Algorithms 8, 9, 10, 11 and 12. The linkable ring signature is similar to the normal ring signature in addition to a link algorithm.

Algorithm 8: Set Up

Input: The security parameter λ.

Output: Public parameter: variable number $n \in \mathbb{N}$, a prime power q and elements $s_0, r_0 \in S$.

1 Choose $n \in \mathbb{N}$ and a prime power q corresponding to the security parameter λ.

2 Randomly sample elements s_0, r_0 from S.

3 **return** *Public parameter:* n, q, s_0, r_0.

Algorithm 9: Linkable key generation

Input: Public parameter n, q, s_0, r_0 and the user i.

Output: Public key for the user i: an element $s_i \in S$.

Private key for the user i: A group element g_i such that $s_i = \alpha(g_i, s_0)$.

1 Randomly sample a group element g_i from G.

2 Compute $s_i \leftarrow \alpha(g_i, s_0)$.

3 **return** *Public key:* s_i. *Private key:* g_i.

Algorithm 10: Link procedure

Input: Two signature $\mathsf{Sig} = (\mathsf{salt}, r, \mathsf{cha}, \mathsf{rsp})$ and $\mathsf{Sig}' = (\mathsf{salt}', r', \mathsf{cha}', \mathsf{rsp}')$.

Output: "Yes" if two signatures are produced by the same secret key. "No" otherwise.

1 **if** $r = r'$ **then**
2 **return** *Yes*
3 **else**
4 **return** *No*

Algorithm 11: Linkable signing procedure

Input: The public key: $s_0, \ldots, s_N$. The private key: g_I. The security parameter λ. The message msg. The commitment scheme $\mathsf{Com} : \{0,1\}^* \to \{0,1\}^\lambda$. A hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\lambda$.

Output: The signature Sig on msg.

1 $r \leftarrow \beta(g_I, r_0)$
2 $\mathsf{com} = (\mathsf{salt}, (\mathsf{com}_i)_{i\in[M]}) \leftarrow \mathcal{P}'_1(s_0, s_1, \ldots, s_N, r)$
3 $\mathsf{cha} \leftarrow \mathcal{H}(\mathsf{msg}||s_1|| \cdots ||s_N||r||\mathsf{com})$
4 $\mathsf{rsp} \leftarrow \mathcal{P}'_2(g_I, I, \mathsf{cha})$
5 **return** $\mathsf{Sig} = (\mathsf{salt}, r, \mathsf{cha}, \mathsf{rsp})$

Algorithm 12: Linkable verification procedure

Input: The public key $s_0, \ldots, s_N \in S$. The signature $\mathsf{Sig} = (\mathsf{salt}, r, \mathsf{cha}, \mathsf{rsp})$. The message msg. A hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\lambda$.

Output: "Yes" if Sig is a valid signature for msg. "No" otherwise.

1 $\mathsf{com} \leftarrow \mathsf{RecoverCom}(s_0, \ldots, s_N, r, \mathsf{salt}, \mathsf{cha}, \mathsf{rsp})$
2 **if** $\mathsf{accept} = \mathcal{V}'_2(\mathsf{com}, \mathsf{cha}, \mathsf{rsp}) \wedge \mathsf{cha} = \mathcal{H}(\mathsf{msg}||s|| \cdots ||s_N||r||\mathsf{com})$ **then**
3 **return** *Yes*
4 **else**
5 **return** *No*

Remark 5.3.5. Since the linkable OR sigma protocol satisfies all properties in Theorem 5.3.4, the correctness, linkability, linkable anonymity and non-frameability of the linkable ring signature in Algorithm 9, 10, 11 and 12 are straightforward.

Remark 5.3.6. Note that our ring signature obtained from OR-Sigma protocol is proven securely only in ROM. As far as we are aware, whether it is secure in QROM is still an open problem.

5.4 An implementation of the ATFE-GMW-FS-BKP ring signature scheme

We implement the GMW-FS-BKP ring signature design based on ATFE. Here, we report the formulas for calculating the parameters, and preliminary experiment results. Some comparisons with known ring signature schemes were presented in Section 1.7.

Some formulas for parameters. Recall that M is the round number, K is the number of non-zero challenges, and C is the number of alternating trilinear forms in each round. To achieve the λ-bits security, we should choose the proper M and K such that $(\frac{M}{K})^K \geqslant 2^\lambda$. We use R to denote the size of the ring. Here we use a trick that evenly divides M rounds into K sections with length of $\lceil \frac{M}{K} \rceil$. For each section, we can construct a seed tree of which the internal seeds are of the size at most $\lambda \cdot \lceil \log_2(\frac{M}{K}) \rceil$.

(1) The public key, private key and signature size of (non-linkable) ring signature in terms of bits are as follows.

$$\text{Public Key Size} = (R+1) \cdot \binom{n}{3} \lceil \log_2 q \rceil,$$

$$\text{Private Key Size} = \binom{n}{3} \lceil \log_2 q \rceil + R \cdot n^2 \lceil \log_2 q \rceil,$$

$$\text{Signature Size} = K(\lambda \cdot \lceil \log_2 \left(\frac{M}{K} \right) \rceil + n^2 \lceil \log_2 q \rceil + 2\lambda \cdot \lceil \log_2 R \rceil + \lambda) + 3\lambda.$$

(2) The public key, private key and signature size of linkable ring signature in terms of bits are as follows.

$$\text{Public Key Size} = (R+1)\cdot\binom{n}{3}\lceil\log_2 q\rceil,$$

$$\text{Private Key Size} = \binom{n}{3}\lceil\log_2 q\rceil + R\cdot n^2\lceil\log_2 q\rceil,$$

$$\text{Signature Size} = K(\lambda\cdot\lceil\log_2\left(\frac{M}{K}\right)\rceil + n^2\lceil\log_2 q\rceil + 2\lambda\cdot\lceil\log_2 R\rceil + \lambda)$$
$$+ 3\lambda + \binom{n}{3}\lceil\log_2 q\rceil.$$

Concrete parameters and reports on the performance. We provide the performance evaluation of our schemes in terms of signature size, as shown in Tables 5.1. Furthermore, Table 5.2 illustrates the signature generation time for our schemes. Our constructions are implemented and measured on a 2.4 GHz Quad-Core Intel Core i5.

Parameters				Size in Bytes				
				R				
n	q	M	K	2^1	2^3	2^6	2^{12}	2^{21}
13	4294967291($\sim 2^{32}$)	850	25	20.5	22.1	24.5	29.3	36.5

Table 5.1: The signature size (KB) of the ring signature. The security meets the NIST level 1.

Parameters				Time in ms						
				R						
n	q	M	K	2^1	2^2	2^3	2^4	2^5	2^6	2^7
13	4294967291($\sim 2^{32}$)	850	25	83	121	205	379	682	1381	2714

Table 5.2: The signing time (ms) of the ring signature. The security meets the NIST level 1.

Part III

Cryptanalysis on MEDS and ALTEQ

Chapter 6

Generic algorithms for MCE and ATFE

6.1 Part preliminaries

Notations. For $n \in \mathbb{N}$, $[n] := \{1, 2, \ldots, n\}$. Let $\mathbb{F}_q$ be the finite field of q elements. We view $\mathbb{F}_q^n$ as the linear space of length-n column vectors over $\mathbb{F}_q$. Let $\mathbb{P} = \mathbb{P}(\mathbb{F}_q^n)$ be the projective space associated with the vector space $\mathbb{F}_q^n$. For a non-zero $\mathbf{u} \in \mathbb{F}_q^n$, we use $\hat{\mathbf{u}} \in \mathbb{P}$ to denote the projective line represented by $\mathbf{u}$. Let $\mathrm{GL}(n, q)$ denote the general linear group of degree n over $\mathbb{F}_q$. We use $\mathrm{M}(m \times n, q)$ to denote the space of $m \times n$ matrices over $\mathbb{F}_q$, and $\mathrm{ATF}(n, q)$ for the space of alternating trilinear forms over $\mathbb{F}_q^n$. For a finite set S, we use $s \leftarrow_R S$ to denote that s is uniformly randomly sampled from S.

Matrix codes and trilinear forms. A trilinear form is a function $\phi : \mathbb{F}_q^m \times \mathbb{F}_q^n \times \mathbb{F}_q^l \to \mathbb{F}_q$ that is linear in each of its three arguments.

Definition 6.1.1 (Trilinear Form Equivalence Problem)**.** Given two trilinear forms $\phi, \psi : \mathbb{F}_q^m \times \mathbb{F}_q^n \times \mathbb{F}_q^l \to \mathbb{F}_q$, the problem asks whether there exists three matrices

$(A, B, C) \in \mathrm{GL}(m, q) \times \mathrm{GL}(n, q) \times \mathrm{GL}(l, q)$, such that for any $(u, v, w) \in \mathbb{F}_q^m \times \mathbb{F}_q^n \times \mathbb{F}_q^l$, $\phi(u, v, w) = \psi(A(u), B(v), C(w))$.

A $[m \times n, l]$-matrix code $\mathcal{C}$ is an l-dimensional subspace of $\mathrm{M}(m \times n, q)$. We defined matrix code equivalence in Definition 1.8.1. Matrix code equivalence reduces to trilinear form equivalence in polynomial time. This is because of the following. Let a matrix code $\mathcal{C}$ be given by an ordered linear basis $(C_1, C_2, \ldots, C_l)$, $C_k \in \mathrm{M}(m \times n, q)$, and $c_{i,j,k}$ denotes the (i, j)-entry of C_k. This gives rise to a trilinear form $\phi_{\mathcal{C}} : \mathbb{F}_q^m \times \mathbb{F}_q^n \times \mathbb{F}_q^l \to \mathbb{F}_q$, that is, $\phi_{\mathcal{C}} = \sum_{i,j,k} c_{i,j,k} u_i v_j w_k$ where $u = (u_1, \ldots, u_m)^t \in \mathbb{F}_q^m$, $v = (v_1, \ldots, v_n)^t \in \mathbb{F}_q^n$, and $w = (w_1, \ldots, w_l)^t \in \mathbb{F}_q^l$. It is straightforward to verify that two matrix codes $\mathcal{C}$ and $\mathcal{D}$ are equivalent if and only if $\phi_{\mathcal{C}}$ and $\phi_{\mathcal{D}}$ are equivalent. Furthermore, if $(A, B, C) \in \mathrm{GL}(m, q) \times \mathrm{GL}(n, q) \times \mathrm{GL}(l, q)$ sends $\phi_{\mathcal{C}}$ ot $\phi_{\mathcal{D}}$, then (A, B) sends $\mathcal{C}$ to $\mathcal{D}$.

We note that the trilinear form equivalence problem differs from the alternating trilinear form equivalence problem, in that three invertible matrices are used in the former, while only one is used in the latter.

Instantiated arguments of trilinear forms. Let $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ be a trilinear form and $\mathbf{u}, \mathbf{v} \in \mathbb{F}_q^n$. We use $\phi(\mathbf{u}, \star, \star)$ to denote the bilinear form obtained by instantiating the first argument of ϕ with $\mathbf{u}$. Let $\phi(\mathbf{u}, \star, \star) = \sum_{j,k} c_{j,k} y_j z_k$ then it has matrix representation $M_{\mathbf{u}} = (c_{j,k})$ with respect to standard basis $e_1, \ldots, e_n$. We use $\phi(\mathbf{u}, \mathbf{v}, \star)$ to denote the linear form obtained by instantiating the first two arguments of ϕ with $\mathbf{u}$ and $\mathbf{v}$, respectively.

Tripartite graphs associated with trilinear forms. Let $\phi \in \mathrm{TF}(\mathbb{F}_q^n)$ be a trilinear form, then we can associate ϕ with a tripartite graph $G_\phi = (U \uplus V \uplus W, E)$ where $U = V = W = \mathbb{P}(\mathbb{F}_q^n)$. To define the edge set E, let $\hat{\mathbf{u}} \in U$, $\hat{\mathbf{v}} \in V$, and $\hat{\mathbf{w}} \in W$. Then $\{\hat{\mathbf{u}}, \hat{\mathbf{v}}\} \in E$, if $\phi(\mathbf{u}, \mathbf{v}, \star)$ is the zero linear form. Similarly, $\{\hat{\mathbf{u}}, \hat{\mathbf{w}}\} \in E$, if $\phi(\mathbf{u}, \star, \mathbf{w})$ is the zero linear form. And $\{\hat{\mathbf{v}}, \hat{\mathbf{w}}\} \in E$, if $\phi(\star, \mathbf{v}, \mathbf{w})$ is the zero linear form.

Rank distribution of random trilinear forms. The following rank distribution of random trilinear forms follows from the well-known fact that the probability of a random matrix in $\mathrm{M}(n, \mathbb{F}_q)$ to be of rank $n-d$ tends to q^{-d^2} as $q \to \infty$ [Bel93, FG15].

Theorem 6.1.2 ([Bel93, FG15]). *Let n, d be positive integers such that $n-d$ is a non-negative number less than n. Then as $q \to \infty$, the average number of projective points with rank $n-d$ of a uniformly random trilinear form $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ tends to q^{-d^2+n-1}.*

Rank distribution of alternating trilinear forms. The following result is due to Beullens [Beu23].

Theorem 6.1.3 ([Beu23, Theorem 2]). *Let n, d be positive integers such that $n-d$ is a non-negative even number less than n. Then as $q \to \infty$, the average number of projective points with rank $n-d$ of a uniformly random alternating trilinear form $\phi \in \mathrm{ATF}(\mathbb{F}_q^n)$ tends to $q^{(-d^2+3d)/2+n-2}$.*

6.2 Finding equivalences of trilinear forms via invariants

We first outline the common framework of our algorithms for ATFE and TFE at a high level, following Beullens (in Section 5.4 of [Beu23]). But in a departure from [Beu23] which relies on invariants derived from graphs on projective points, we design new global invariants. The invariant functions for ATF and TF will be of the form

$$F_0 : \mathrm{TF}(\mathbb{F}_q^n) \times \mathbb{P}(\mathbb{F}_q^n) \to X_0,$$
$$F_1 : \mathrm{ATF}(\mathbb{F}_q^n) \times \mathbb{P}(\mathbb{F}_q^n) \to X_1$$

and explicitly constructed in the following sections. The subscript 0 in the function and the target set indicates that it is associated with TF. Likewise, the subscript 1 indicates an association with ATF. We will provide the detailed algorithm for MCE and ATFE in Chapter 7 and Chapter 8 respectively.

Invariants. To illustrate the notion of invariants, let us first name the actions underlying MCE and ATFE in the language of trilinear forms.

Definition 6.2.1 (MCE Action)**.** For a trilinear form $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \longrightarrow \mathbb{F}_q$ and a triple of matrices $(A, B, C) \in \mathrm{GL}(n, q)^3$, define the trilinear form

$$\phi_{A,B,C} : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \longrightarrow \mathbb{F}_q$$
$$(x, y, z) \longmapsto \phi(Ax, By, Cz).$$

We design F_0 as a pairing of the trilinear form and the projective space that is invariant under twisting the trilinear form and the projective space. The trilinear form is twisted by the $\mathrm{GL}(n, q)^3$ MCE Action. The projective space is twisted by the inverse of the matrix acting on the first dimension of the trilinear form. Formally, the invariant for MCE action needs to satisfy that

$$\forall \phi \in \mathrm{TF}(\mathbb{F}_q^n), \forall \hat{\mathbf{v}} \in \mathbb{P}(\mathbb{F}_q^n), \forall (A, B, C) \in \mathrm{GL}(n, q)^3, F_0(\phi, \hat{\mathbf{v}}) = F_0(\phi_{A,B,C}, A^{-1}\hat{\mathbf{v}}).$$

Definition 6.2.2 (ATFE Action)**.** For a trilinear form $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \longrightarrow \mathbb{F}_q$ and a matrix $A \in \mathrm{GL}(n, q)$, define the trilinear form

$$\phi_A : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \longrightarrow \mathbb{F}_q$$
$$(x, y, z) \longmapsto \phi(Ax, Ay, Az).$$

We design the function F_1 as a pairing of the trilinear form and the projective space that is invariant under twisting the trilinear form by the ATFE action and the projective space by the inverse of the matrix defining the ATFE action. Formally,

$$\forall \phi \in \mathrm{ATF}(\mathbb{F}_q^n), \forall \hat{\mathbf{v}} \in \mathbb{P}(\mathbb{F}_q^n), \forall A \in \mathrm{GL}(n, q), F_1(\phi, \hat{\mathbf{v}}) = F_1(\phi_A, A^{-1}\hat{\mathbf{v}}).$$

Distinguishing invariant. The invariant function F_0 is called distinguishing if for all $\phi \in \mathrm{TF}(\mathbb{F}_q^n)$,

$$\Pr_{(\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2) \leftarrow_R \mathbb{P}(\mathbb{F}_q^n)^2} (F_0(\phi, \hat{\mathbf{v}}_1) \neq F_0(\phi, \hat{\mathbf{v}}_2)) \approx 1.$$

We will specify the meaning of ≈ 1 in the following. Likewise, F_1 is called distinguishing if for all $\phi \in \mathrm{ATF}(\mathbb{F}_q^n)$,

$$\Pr_{(\hat{\mathbf{v}}_1,\hat{\mathbf{v}}_2)\leftarrow_R \mathbb{P}(\mathbb{F}_q^n)^2} (F_1(\phi,\hat{\mathbf{v}}_1) \neq F_1(\phi,\hat{\mathbf{v}}_2)) \approx 1.$$

An algorithm template based on distinguishing invariants. With such distinguishing invariant functions at hand, we have the following generic algorithm for MCE and ATFE. The version for ATFE is specified in parentheses.

To start with, recall that for a trilinear form $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ and $\mathbf{v} \in \mathbb{F}_q^n$, the rank of $\phi(\mathbf{v},\star,\star)$ (see Section 2.2) is an invariant, which has been utilised in [BFV13,Beu23]. Also note that $\mathrm{rk}(\phi(\mathbf{v},\star,\star)) = \mathrm{rk}(\phi(\lambda\mathbf{v},\star,\star))$ for non-zero $\lambda \in \mathbb{F}_q$, so we can talk about the rank of $\phi(\hat{\mathbf{v}},\star,\star)$ for $\hat{\mathbf{v}} \in \mathbb{P}(\mathbb{F}_q^n)$.

This rank invariant cannot be distinguished. Still, the new invariants considered in this thesis are further refinements of the rank invariant, as will be seen below. In particular, the generic algorithm is parametrized by this rank R, which would be specified later depending on the specific invariants.

Input: Two equivalent (alternating) trilinear forms $\phi, \psi \in \mathrm{TF}(\mathbb{F}_q^n)$(or $\mathrm{ATF}(\mathbb{F}_q^n)$).

Output: $A, B, C \in \mathrm{GL}(n,q)$ such that $\phi_{A,B,C} = \psi$ (or $A \in \mathrm{GL}(n,q)$ such that $\phi_A = \psi$).

Algorithm **(1)** Pick a positive number $R \leqslant n$. Let

$$\mathbb{P}_{\phi,R} := \left\{\hat{\mathbf{v}} \in \mathbb{P}(\mathbb{F}_q^n) \mid \mathrm{rk}(\phi(\hat{\mathbf{v}},*,*)) = R\right\},$$

$$\mathbb{P}_{\psi,R} := \left\{\hat{\mathbf{v}} \in \mathbb{P}(\mathbb{F}_q^n) \mid \mathrm{rk}(\psi(\hat{\mathbf{v}},*,*)) = R\right\}$$

denote the respective set of points where the trilinear forms specialize in the first dimension to give rank R matrices. Independently sample a set $L_{\phi,R}$ of $\sqrt{|\mathbb{P}_{\phi,R}|}$ points from $\mathbb{P}_{\phi,R}$ and a set $L_{\psi,R}$ of $\sqrt{|\mathbb{P}_{\psi,R}|}$ points from $\mathbb{P}_{\psi,R}$. Since ϕ and ψ are isomorphic, $\mathbb{P}_{\phi,R} = A \times \mathbb{P}_{\psi,R}$ and we denote their cardinality as $N_R := \|\mathbb{P}_{\phi,R}\| = \|\mathbb{P}_{\psi,R}\|$. Therefore $L_{\phi,R}$ and $L_{\psi,R}$ are both $\sqrt{N_R}$-sized subsets of the same set of size N_R.

(2) Apply the invariant function F_i (where $i = 0$ for MCE and $i = 1$ for ATFE) to each element in $L_{\phi,R}$ and $L_{\psi,R}$. Find a pair $(\hat{\mathbf{v}}, \hat{\mathbf{v}'})$ for which $F_i(\phi, \hat{\mathbf{v}}) = F_i(\psi, \hat{\mathbf{v}'})$, where $\hat{\mathbf{v}} \in L_{\phi,R}$ and $\hat{\mathbf{v}'} \in L_{\psi,R}$. The existence of such a pair is likely due to the birthday paradox.

(3) For MCE, such a pair reveals the desired output $(A, B, C) \in \mathrm{GL}(n, q)^3$ through linear algebra, as we describe in Chapter 7. To solve the ATFE, feed the matching pair $(\hat{\mathbf{v}}, \hat{\mathbf{v}'})$ as the partial information into the Gröbner basis computation in [TDJ+22, BBC+20]. This Gröbner basis computation is a heuristic that finds in polynomial time an $A \in \mathrm{GL}(n, q)$ (if it exists) such that $\phi_A = \psi$ and $A^{-1}\hat{\mathbf{v}} = \hat{\mathbf{v}'}$.

The complexity of the above algorithm parameterized by the target rank R can be estimated as

$$O\left(\sqrt{N_R} \cdot (\text{samp-cost} + \text{inv-cost}) + \text{recover-cost}\right). \tag{6.1}$$

The sampling cost samp-cost refers to the cost of sampling a rank-R (projective) point, that is, a point in $\mathbb{P}_{\phi,R}$ (or equivalently in $\mathbb{P}_{\psi,R}$). And inv-cost denotes the cost of invariant computation for each point. The cost of recovering the isomorphism given a collision is denoted by recover-cost. Also note that for the invariant to be distinguishing enough in the above procedure, we need to have $\Pr_{(\hat{\mathbf{v}}_1,\hat{\mathbf{v}}_2)\leftarrow_R \mathbb{P}(\mathbb{F}_q^n)^2}(F_0(\phi, \hat{\mathbf{v}}_1) = F_0(\phi, \hat{\mathbf{v}}_2)) = O(1/N_R)$.

In the following two chapters, we describe algorithms in this general framework tailored to MCE and ATFE, by describing the invariant functions and optimizing the rank R.

Chapter 7

Algorithms for matrix code equivalence

In this chapter, we introduce an algorithm for the matrix code (or trilinear form) equivalence problem. Specifically, given two trilinear forms $\phi \in \mathrm{TF}(\mathbb{F}_q^n)$ and $\psi \in \mathrm{TF}(\mathbb{F}_q^n)$ that are equivalent, the algorithm computes an equivalence $(A, B, C) \in \mathrm{GL}(n, q) \times \mathrm{GL}(n, q) \times \mathrm{GL}(n, q)$ between ϕ and ψ. The algorithm runs in time $O(q^{(n-2)/2} \cdot (q \cdot n^3 + n^4) \cdot (\log(q))^2)$.

Main idea. To instantiate the algorithm outlined in Section 6.2, the primary bottleneck is identifying invariants with sufficient distinguishing power. The main idea of the algorithm is to associate distinguishing invariants to corank-1 points, specifically for those $\hat{\mathbf{u}} \in \mathbb{P}(\mathbb{F}_q^n)$ such that the bilinear form $\phi(\mathbf{u}, \star, \star)$ is of rank $n - 1$. We shall occasionally call such projective lines as corank-1 points. Recall there is a tripartite graph $G_\phi = (U \uplus V \uplus W, E)$ associated with ϕ where $U = V = W = \mathbb{P}(\mathbb{F}_q^n)$. Each corank-1 point $\hat{\mathbf{u}} \in U$ has a unique neighbor $\hat{\mathbf{v}} \in V$, namely the one dimensional left kernel of the bilinear form $\phi(\mathbf{u}, \star, \star)$. Since $\phi(\star, \mathbf{v}, \star)$ has $\mathbf{u}$ in its left kernel, $\phi(\star, \mathbf{v}, \star)$ has co-rank at least 1. If $\phi(\star, \mathbf{v}, \star)$ is of corank-1, it has a unique neighbour $\hat{\mathbf{w}} \in W$. Repeating this procedure leads to a path on G_ϕ. We continue building this path until

reaching length $3n$, collecting n points each from U, V and W. Such a path is built without ambiguity if and only if at every iteration we get a point of corank-1.

Our experiments show that for most starting points $\hat{\mathbf{u}}$, we do obtain a path of length $3n$ without ambiguity and that the vector n-tuples collected in each of the sets U, V and W are linearly independent respectively. We use these three vector tuples to transform ϕ to $\tilde{\phi}[\mathbf{u}]$ which depends only on the vectors on this path.

To make this an isomorphism invariant indexed with $\hat{\mathbf{u}}$ (instead of with $\mathbf{u}$), we need to remove the ambiguity caused by the scalar multiples, which can be done easily by locating non-zero evaluations of $\tilde{\phi}[\mathbf{u}]$ on about $3n$ inputs of the form (e_i, e_j, e_k). This gives us $\bar{\phi}[\hat{\mathbf{u}}]$ which is an invariant associated with $\hat{\mathbf{u}}$. Our experiments show that this invariant is distinguishing, i.e. different $\hat{\mathbf{u}}$ results in different $\bar{\phi}[\hat{\mathbf{u}}]$. This allows for an application of the birthday algorithm.

It is known from Theorem 6.1.2 that for a random ϕ, there exist approximately q^{n-2} corank-1 points. Thus we get an algorithm running in time $O((q^{(n/2)} + q^{(n-2)/2}) \cdot \text{poly}(n, q))$ by instantiating the above invariant.

7.1 From a vector to three vector tuples

Corank-1 points of trilinear forms and paths on G_ϕ. Suppose a non-zero $\mathbf{u}_1 \in \mathbb{F}_q^n$ satisfies that $\phi(\mathbf{u}_1, \star, \star)$ is of corank-1 as a bilinear form. Consider the following steps.

(1) As $\phi(\mathbf{u}_1, \star, \star)$ is of corank-1, there exists a unique $\hat{\mathbf{v}_1} \in \mathbb{P}$ such that $\phi(\mathbf{u}_1, \mathbf{v}_1, \star)$ is the zero linear form.

(2) If $\phi(\star, \mathbf{v}_1, \star)$ is of corank-1, then there exists a unique $\hat{\mathbf{w}_1} \in \mathbb{P}$, such that $\phi(\star, \mathbf{v}_1, \mathbf{w}_1)$ is the zero linear form.

(3) If $\phi(\star, \star, \mathbf{w}_1)$ is of corank-1, then there exists a unique $\hat{\mathbf{u}_2} \in \mathbb{P}$, such that $\phi(\mathbf{u}_2, \star, \mathbf{w}_1)$ is the zero linear form.

If $\hat{\mathbf{u}}_1 \neq \hat{\mathbf{u}}_2$, then the above procedure produces a path $(\hat{\mathbf{u}}_1, \hat{\mathbf{v}}_1, \hat{\mathbf{w}}_1, \hat{\mathbf{u}}_2)$ in $G(\phi)$. We can continue the above procedure as follows.

(1) Let $L_U = (u_1)$, $L_V = ()$, and $L_W = ()$.

(2) For $i = 1$ to n, do the following:

a) Compute the unique $\hat{\mathbf{v}}_i \in \mathbb{P}(\mathbb{F}_q^n)$, such that $\phi(\mathbf{u}_i, \mathbf{v}_i, \star) = 0$.

b) If the corank of $\phi(\star, \mathbf{v}_i, \star)$ is not 1, or if $\mathbf{v}_i \in \text{span}(L_V)$, terminate and report "Fail". Otherwise, add $\mathbf{v}_i$ to L_V.

c) Compute the unique $\hat{\mathbf{w}}_i \in \mathbb{P}(\mathbb{F}_q^n)$, such that $\phi(\star, \mathbf{v}_i, \mathbf{w}_i) = 0$.

d) If the corank of $\phi(\star, \star, \mathbf{w}_i)$ is not 1, or if $\mathbf{w}_i \in \text{span}(L_W)$, terminate and report "Fail". Otherwise, add $\mathbf{w}_i$ to L_W.

e) If $i = n$, break.

f) Compute the unique $\hat{\mathbf{u}_{i+1}} \in \mathbb{P}(\mathbb{F}_q^n)$, such that $\phi(\mathbf{u}_{i+1}, \star, \mathbf{w}_i) = 0$.

g) If the corank of $\phi(\mathbf{u}_{i+1}, \star, \star)$ is not 1, or if $\mathbf{u}_{i+1} \in \text{span}(L_U)$, terminate and report "Fail". Otherwise, add $\mathbf{u}_{i+1}$ to L_U.

If the above procedure does not return "Fail", then we obtain three vector tuples $L_U = (\mathbf{u}_1, \ldots, \mathbf{u}_n)$, $L_V = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$, and $L_W = (\mathbf{w}_1, \ldots, \mathbf{w}_n)$, such that $\mathbf{u}_i$'s (resp, $\mathbf{v}_i$'s, $\mathbf{w}_i$'s) are linearly independent.

7.2 Corank-1 invariants from three vector tuples

Suppose that starting from a corank-1 $\mathbf{u}_1 \in \mathbb{F}_q^n$, we obtain three vector tuples L_U, L_V, and L_W, which are canonically associated with $\mathbf{u}_1$. We then treat L_U, L_V, and L_W as invertible matrices, that is, $L_U = \begin{bmatrix} \mathbf{u}_1 & \ldots & \mathbf{u}_n \end{bmatrix}^t$. Define a trilinear form $\tilde{\phi} : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ by $\tilde{\phi}(x, y, z) = \phi(L_U(x), L_V(y), L_W(z))$. This $\tilde{\phi}$ is almost an isomorphism invariant associated with $\mathbf{u}_1$ – almost because there is an ambiguity associated with the representing vectors of $\hat{\mathbf{u}}_i$, $\hat{\mathbf{v}}_j$, and $\hat{\mathbf{w}}_k$.

To remove this ambiguity, we need to study the canonical form of $\tilde{\phi}$ under the action of $\mathrm{D}(n,q) \times \mathrm{D}(n,q) \times \mathrm{D}(n,q)$, where $\mathrm{D}(n,q)$ denotes the group of invertible diagonal $n \times n$ matrices over $\mathbb{F}_q$.

This can be done relatively easily when, for any $i, j, k \geqslant 3$,

$$\begin{aligned} &a_i := \tilde{\phi}(e_i, e_2, e_1), b_j := \tilde{\phi}(e_1, e_j, e_1), c_k := \tilde{\phi}(e_1, e_2, e_k), d_1 := \tilde{\phi}(e_1, e_2, e_1), \\ &d_2 := \tilde{\phi}(e_2, e_3, e_5), d_3 := \tilde{\phi}(e_1, e_3, e_2), d_4 := \tilde{\phi}(e_2, e_1, e_2) \text{ are non-zero.} \end{aligned} \tag{7.1}$$

In this case, we can use the action of $\mathrm{D}(n,q) \times \mathrm{D}(n,q) \times \mathrm{D}(n,q)$ to set $a_i, b_j, c_k, d_1, d_2, d_3$ and d_4 to be 1. More specifically, let $(F, G, H) \in \mathrm{D}(n,q) \times \mathrm{D}(n,q) \times \mathrm{D}(n,q)$, where $F = \mathrm{diag}(f_1, \ldots, f_n)$, $G = \mathrm{diag}(g_1, \ldots, g_n)$, and $H = \mathrm{diag}(h_1, \ldots, h_n)$. Then set f_i, g_j, and h_k to satisfy that, for $3 \leq i, j, k \leq n$,

$$\begin{aligned} &f_1 g_2 h_1 = 1/d_1, f_i/f_1 = d_1/a_i, g_j/g_2 = d_1/b_j, h_k/h_1 = d_1/c_k, \\ &f_2 = 1/(g_3 h_5 d_2), h_2 = 1/(f_1 g_3 d_3), g_1 = 1/(f_2 h_2 d_4). \end{aligned} \tag{7.2}$$

Let $\bar{\phi} : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ be defined by $\bar{\phi}(x, y, z) = \tilde{\phi}(F(x), G(y), H(z))$. Then $\bar{\phi}(e_i, e_j, e_k) = f_i g_j h_k \tilde{\phi}(e_i, e_j, e_k)$. Therefore,

$$\bar{\phi}(e_1, e_2, e_1) = f_1 g_2 h_1 \tilde{\phi}(e_1, e_2, e_1) = 1/d_1 \cdot d_1 = 1.$$

For $i \geq 3$,

$$\begin{aligned} & \bar{\phi}(e_i, e_2, e_1) \\ = \;& f_i g_2 h_1 \tilde{\phi}(e_i, e_2, e_1) \\ = \;& (f_i/f_1) f_1 g_2 h_1 \tilde{\phi}(e_i, e_2, e_1) \\ = \;& (d_1/a_i) \cdot (1/d_1) \cdot a_i = 1. \end{aligned}$$

Similarly, it can be verified that $\bar{\phi}(e_1, e_j, e_1) = \bar{\phi}(e_1, e_2, e_k) = 1$ for $j, k \geq 3$. Additionally, we can verify that $\bar{\phi}(e_1, e_2, e_1) = \bar{\phi}(e_2, e_3, e_5) = \bar{\phi}(e_2, e_1, e_2) = \bar{\phi}(e_1, e_3, e_2) = 1$. Furthermore, for any $i, j, k \geqslant 3$,

$$\bar{\phi}(e_i, e_j, e_k)$$

$$\begin{aligned} &= f_i g_j h_k \tilde{\phi}(e_i, e_j, e_k) \\ &= (f_i/f_1)(g_j/g_2)(h_k/h_1) f_1 g_2 h_1 \tilde{\phi}(e_i, e_j, e_k) \\ &= \frac{d_1^4 \tilde{\phi}(e_i, e_j, e_k)}{a_i b_j c_k}; \end{aligned}$$

for $i = 2$ and any $j, k \geqslant 3$,

$$\begin{aligned} &\bar{\phi}(e_2, e_j, e_k) \\ &= f_2 g_j h_k \tilde{\phi}(e_2, e_j, e_k) \\ &= (f_2/f_1)(g_j/g_2)(h_k/h_1) f_1 g_2 h_1 \tilde{\phi}(e_2, e_j, e_k) \\ &= \frac{b_3 b_5 \tilde{\phi}(e_2, e_j, e_k)}{d_1 d_2 b_j c_k}; \end{aligned}$$

for $k = 2$ and any $i, j \geqslant 3$,

$$\begin{aligned} &\bar{\phi}(e_i, e_j, e_2) \\ &= f_i g_j h_2 \tilde{\phi}(e_i, e_j, e_2) \\ &= (f_i/f_1)(g_j/g_2)(h_2/h_1) f_1 g_2 h_1 \tilde{\phi}(e_i, e_j, e_2) \\ &= \frac{b_3 \tilde{\phi}(e_i, e_j, e_2)}{d_3 a_i b_j}; \end{aligned}$$

for $j = 1$ and any $i, k \geqslant 3$,

$$\begin{aligned} &\bar{\phi}(e_i, e_1, e_k) \\ &= f_i g_1 h_k \tilde{\phi}(e_i, e_1, e_k) \\ &= (f_i/f_1)(g_1/g_2)(h_k/h_1) f_1 g_2 h_1 \tilde{\phi}(e_i, e_1, e_k) \\ &= \frac{d_1^7 d_2 d_3 d_4 \tilde{\phi}(e_i, e_1, e_k)}{b_3^2 b_5 a_i c_k}; \end{aligned}$$

So $\bar{\phi}$ is completely determined by the conditions in Equation 7.2.

The above suggests that $\bar{\phi}[\hat{\mathbf{u}}_1] := \bar{\phi}$ is an isomorphism invariant associated with $\hat{\mathbf{u}}_1 \in \mathbb{P}(\mathbb{F}_q^n)$, assuming that $\tilde{\phi}$ satisfies Equation 7.1.

7.3 Description of the algorithm

Given the above preparations, the algorithm works as follows.

Input two equivalent trilinear forms $\phi, \psi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$.

Output an equivalence $(A, B, C) \in \mathrm{GL}(n, q) \times \mathrm{GL}(n, q) \times \mathrm{GL}(n, q)$.

Algorithm **(1)** For ϕ, construct a list S_ϕ of $q^{(n-2)/2}$ corank-1 $\hat{\mathbf{u}} \in \mathbb{P}$ together with the isomorphism invariant $\bar{\phi}[\hat{\mathbf{u}}]$ as follows.

a) Compute one corank-1 $\hat{\mathbf{u}} \in \mathbb{P}$ by sampling randomly $\mathbf{u} \in \mathbb{F}_q^n$ q times.

b) For $\hat{\mathbf{u}} \in \mathbb{P}$, compute three vector tuples L_U, L_V, and L_W as in Section 7.1.

c) Use L_U, L_V and L_W to transform ϕ to $\tilde{\phi}[\mathbf{u}]$.

d) Use the method in Section 7.2 to transform $\tilde{\phi}[\mathbf{u}]$ to $\bar{\phi}[\hat{\mathbf{u}}]$.

(2) For ψ, construct a list S_ψ of $q^{(n-2)/2}$ corank-1 $\hat{\mathbf{u}} \in \mathbb{P}(\mathbb{F}_q^n)$ together with the isomorphism invariant $\bar{\psi}[\hat{\mathbf{u}}]$ as above.

(3) Find $\hat{\mathbf{u}}$ from S_ϕ, and $\hat{\mathbf{u}'}$ from S_ψ, such that $\bar{\phi}[\hat{\mathbf{u}}]$ and $\bar{\psi}[\hat{\mathbf{u}'}]$ are the same.

(4) An equivalence (A, B, C) from ϕ to ψ can be obtained by composing the transformations from ϕ to $\bar{\phi}[\hat{\mathbf{u}}]$ and from ψ to $\bar{\psi}[\hat{\mathbf{u}'}]$.

Time analysis of the above algorithm. We assume that the modular arithmetic complexity in $\mathbb{F}_q$ is in time $O((\log q)^2)$, and the number of arithmetic operations for $n \times n$ matrix computations (such as matrix multiplication and rank computation) is $O(n^3)$. As in the practical setting, n is small and matrices are dense, this should be a reasonable estimate (rather than using $O(n^\omega)$ where ω is the matrix multiplication exponent).

Step 1 is a For-loop contributing a multiplicative factor of $q^{(n-2)/2}$ to steps (a) to (d). Step (a) samples vectors in $\mathbb{F}_q^n$ and computes the ranks of the associated matrices for q times, so its complexity is $O(q \cdot (n \cdot \log(q) + n^3 \cdot (\log q)^2))$. Step (b) constructs three n-tuples of vectors. Each vector in this n-tuple is obtained by solving a system of n linear equations in n variables. So Step (b) costs $O(n \cdot n^3 \cdot (\log q)^2) = O(n^4 \cdot (\log q)^2)$. Step (c) requires $3n$ $n \times n$ matrix multiplications, so its complexity is also $O(n^4 \cdot (\log q)^2)$. For Step (d), the method in Section 7.2 takes $O(n^3 \cdot (\log q)^2)$ time.

Taking into account of the For-loop factor, the total cost for steps 1 and (a) to (d) is $O(q^{(n-2)/2} \cdot (q \cdot n^3 + n^4) \cdot (\log(q))^2)$.

Once the two lists are constructed, finding a collision and using that to construct an isomorphism takes time $O(\log(q^{(n-2)/2}))$ as we can assume that the lists S_ϕ and S_ψ are sorted. Therefore steps 2 to 4 contribute to a running time of lower order, and the running time of the whole algorithm is $O(q^{(n-2)/2} \cdot (q \cdot n^3 + n^4) \cdot (\log(q))^2)$.

Correctness analysis of the above algorithm. We assume that $\bar{\phi}[\hat{\mathbf{u}}]$ is a distinguishing invariant of $\hat{\mathbf{u}}$. Then by birthday paradox, the above algorithm returns $\hat{\mathbf{u}}$ from S_ϕ, and $\hat{\mathbf{u}'}$ from S_ψ, such that $\bar{\phi}[\hat{\mathbf{u}}]$ and $\bar{\psi}[\hat{\mathbf{u}'}]$ are the same, with constant probability.

7.4 Heuristic assumptions for the invariant

We now reflect on several assumptions required for using $\bar{\phi}[\mathbf{u}_1]$ for $\mathbf{u}_1 \in \mathbb{F}_q^n$ with $\phi(\mathbf{u}_1, \star, \star)$ being of corank-1.

(1) We assume that we can obtain three vector tuples L_U, L_V, L_W.

(2) We assume that $\tilde{\phi}$, the trilinear form obtained after applying L_U, L_V, and L_W, satisfies Equation 7.1.

(3) We assume that the corank-1 invariant $\bar{\phi}[\mathbf{u}_1]$ is distinguishing.

We next argue in favor of each of these heuristics.

Heuristic 1. To build the vector tuples L_U, L_V, and L_W, it suffices (1) to perform a walk with corank-1 points for $3n$ successful steps, and (2) the vectors in L_U (resp. L_V, L_W) be linearly independent.

We argue for (1), by making the same assumption as in Beullens' algorithms [Beu23], namely those points along such a walk are close to independent randomly sampled. In particular, the probability of getting a walk with corank-1 points for $3n$

steps can be estimated as follows. The probability of a corank-1 point having a corank-2 neighbor is asymptotically $O(1/q^2)$ [Beu23]. Therefore, the probability of walking for $3n$ steps with corank-1 points is lower bounded by $1 - O(n/q^2)$, assuming points along such a walk are close to independent randomly sampled.

We argue for (2) using algebraic-geometry. To this end, consider a generic starting corank-1 vector $\mathbf{u}_1$ and think of its coordinate vector $(\mathbf{u}_{1,1}, \mathbf{u}_{1,2}, \ldots, \mathbf{u}_{1,n})$ as n indeterminates. The corank-1 assumption implies that there is a unique projective $\hat{\mathbf{v}}_1$ such that $\phi(\mathbf{u}_1, \mathbf{v}_1, *) = 0$ (that is, the zero dual vector). The coordinates of $\mathbf{v}_1$ can be expressed as some vector of polynomials in the coordinate ring of $\mathbf{u}_1$, for instance using the adjugate matrix of $\phi(\mathbf{u}_1, *, *)$. Call this vector of polynomials as $(f^{\phi}_{\mathbf{v}_1,j})_{1\leqslant j\leqslant n} \in \left(\mathbb{F}_q[\mathbf{u}_{1,1}, \mathbf{u}_{1,2}, \ldots, \mathbf{u}_{1,n}]\right)^n$. The superscript ϕ signifies that the coefficients of each $f^{\phi}_{\mathbf{v}_1,j}$ depend only on the tensor ϕ. Repeating a similar process starting with the coordinate vector $(f^{\phi}_{\mathbf{v}_1,j})_{1\leqslant j\leqslant n}$ of $\mathbf{v}_1$, we obtain the coordinates $(f^{\phi}_{\mathbf{w}_1,j})_{1\leqslant j\leqslant n} \in \left(\mathbb{F}_q[\mathbf{u}_{1,1}, \mathbf{u}_{1,2}, \ldots, \mathbf{u}_{1,n}]\right)^n$ of $\mathbf{w}_1 \in L_W$. Note that each coordinate is a polynomial in the coordinate ring of the generic starting vector $\mathbf{u}_1$. Continuing this way, we can express each element of L_U, L_V, and L_W as a vector of polynomials in the co-ordinate ring of $\mathbf{u}_1$. The vectors in L_U being linearly independent can be expressed as a polynomial condition on the coordinates of $\mathbf{u}_1$, namely the determinant of the matrix $(f^{\phi}_{\mathbf{u},j})_{u\in L_U,1\leqslant j\leqslant n}$ vanishing. In particular, the variety of dependent L_U has co-dimension at least one, as long as the symbolic determinant $\det\left((f^{\phi}_{\mathbf{u},j})_{u\in L_U,1\leqslant j\leqslant n}\right)$ is not identically zero. The matrix $(f^{\phi}_{\mathbf{u},j})_{u\in L_U,1\leqslant j\leqslant n}$ depends only on ϕ. For the random choice of ϕ induced by key generation, the symbolic determinant $\det\left((f^{\phi}_{\mathbf{u},j})_{u\in L_U,1\leqslant j\leqslant n}\right)$ is almost certainly not identically zero. Therefore, its roots, which constitute the pathological variety of dependent L_U have co-dimension at least one. Therefore with probability at least $1 - 1/q$, we expect the co-ordinates of a random starting vector $\mathbf{u}_1$ to not be in this variety, implying that the L_U vectors are linearly independent. The probability $1 - 1/q$ is only a crude estimate. For a precise bound taking into account the structure of the polynomial, we can invoke the Schwartz–Zippel lemma or more generally the Lang–Weil bound. The Lang–Weil bound subsumes the Schwartz–Zippel lemma and gives stronger bounds in

many cases where more (such as number of irreducible components, degree, smoothness, etc.) is known about the polynomial $\det\left((f^{\phi}_{\mathbf{u},j})_{u\in L_U, 1\leqslant j\leqslant n}\right)$. In either case, to unconditionally prove that a random $\mathbf{u}_1$ is not in this variety, it helps if the degree of the polynomial is not too big. Naively, the polynomial produced through expansion is of exponential degree, but this is unlikely to be optimal, as shown in the experiment part. We leave unconditional proof of the validity of this heuristic to future work.

Heuristic 2. Here we assume that $O(n)$ entries in the transformed tensor are non-zero. Therefore, the probability of this assumption failing increases as q decreases and n increases. Note that this assumption is used only to deal with diagonal group actions, and more specialized techniques can be done to reduce the failure probability of this step.

Heuristic 3. We prove that the invariants generated by our algorithm are distinguishing with high probability, under the following well-studied conjecture from [RST24], which we re-phrase in tensor notation. To this end, define the automorphism group of a tensor $\phi \in TF(\mathbb{F}_q)$ as the subgroup $\mathrm{Aut}(\phi) \leqslant \mathrm{GL}(n,q)^3$ such that

$$\forall (A,B,C) \in \mathrm{Aut}(\phi), \forall (x,y,z) \in \mathbb{F}_q^n, \phi(Ax,By,Cz) = \phi(x,y,z).$$

Clearly, scalar matrices of the form

$$\{(\lambda I_n, \mu I_n, \nu I_n) \mid \lambda\mu\nu = 1, (\lambda,\mu,\nu) \in \left(\mathbb{F}_q^{\times}\right)^3\} \leqslant \mathrm{Aut}(\phi)$$

form a subgroup of the automorphism group. We say that the automorphism group $\mathrm{Aut}(\phi)$ is trivial or equivalently that ϕ has trivial automorphism group if and only if

$$\{(\lambda I_n, \mu I_n, \nu I_n) \mid \lambda\mu\nu = 1, (\lambda,\mu,\nu) \in \left(\mathbb{F}_q^{\times}\right)^3\} = \mathrm{Aut}(\phi).$$

That is, all automorphisms are merely triples of scalar matrices.

Conjecture 7.4.1. For uniformly random $\phi \in TF(\mathbb{F}_q^n)$, with probability negligibly close to 1, the automorphism group $\mathrm{Aut}(\phi)$ is trivial.

This conjecture is stated as a "mild assumption" in [RST24], where the authors provide convincing theoretic and empirical evidence. In fact, this conjecture is assumed true in half of the complexity theoretic reductions in the web of problems centered around MCE ([RST24, Fig. 1]), that lay as the foundation for MEDS.

Consider the corank-1 invariant $\bar{\phi}[\hat{\mathbf{u}}]$ constructed at a successful completion of the first step of the algorithm. We prove in the subsequent lemma 7.4.2 that $\bar{\phi}[\hat{\mathbf{u}}]$ is distinguishing if the isomorphism class of ϕ has a trivial automorphism group.

Lemma 7.4.2. *If $\phi \in TF(\mathbb{F}_q^n)$ has the trivial automorphism group, then the isomorphism invariant $(\phi, \hat{\mathbf{u}}) \longmapsto \bar{\phi}[\hat{\mathbf{u}}]$ determined by step 1 of the algorithm is distinguishing.*

Proof. Recall the notation in the description of the algorithm, to aid in the proof sketch. Let (L_U, L_V, L_W) and (L'_U, L'_V, L'_W) be the two vector tuples produced starting from different $\mathbf{u}$ and $\mathbf{u}'$, respectively. Let $\bar{\phi}[\hat{\mathbf{u}}]$ and $\bar{\phi}[\hat{\mathbf{u}'}]$ respectively denote the images of the invariant computed by step 1 of the algorithm. If the algorithm samples two $\bar{\phi}[\hat{\mathbf{u}}]$ and $\bar{\phi}[\hat{\mathbf{u}'}]$ that are the same, then the respective vector tuples (L_U, L_V, L_W) and (L'_U, L'_V, L'_W) can be composed to get a non-trivial automorphism in $\mathrm{Aut}(\phi)$. But $\phi \in TF(\mathbb{F}_q^n)$ has the trivial automorphism group, therefore $\bar{\phi}[\hat{\mathbf{u}}]$ and $\bar{\phi}[\hat{\mathbf{u}'}]$ are distinct, implying the invariant is distinguishing. □

The MEDS key generation algorithm chooses a ϕ uniformly at random from $TF(\mathbb{F}_q^n)$. Assuming conjecture 7.4.1, $\mathrm{Aut}(\phi)$ is trivial with probability negligibly close to 1. Therefore, lemma 7.4.2 applies in our setting (except possibly with negligibly small probability), implying $(\phi, \bar{\mathbf{u}}) \longmapsto \bar{\phi}[\hat{\mathbf{u}}]$ is distinguishing.

Experimental support. We carry out experiments on Magma [BJP97] for $n = 6$ to 10 and $q = 1021$ to verify the assumptions as above.

We examine Assumptions 1, 2, and 3 sequentially as follows. That is, for a point $\mathbf{u}$, we first verify if assumption 1 holds. If so, then we check if assumption 2 holds for $\mathbf{u}$. If both assumptions 1 and 2 hold, we call $\mathbf{u}$ an *effective point.* In Table 7.1, we sample

q \ n	6	7	8	9	10	11	12	13	14
509	7/26/967	1/39/960	5/40/955	5/41/954	1/70/929	12/58/930	6/57/937	11/67/922	5/81/914
1021	8/10/982	5/16/979	10/20/970	4/28/968	2/18/980	1/27/972	3/31/966	2/30/968	1/29/970
2039	1/13/986	1/13/986	3/14/983	2/8/990	0/18/982	0/18/982	1/15/984	2/17/981	0/18/982
4093	1/5/994	1/7/992	1/5/994	1/7/992	0/6/994	2/6/992	0/13/987	2/11/987	0/10/990
8191	0/3/997	0/2/998	1/2/997	0/2/998	1/4/995	0/3/997	0/5/995	1/8/991	1/5/994
16381	0/0/1000	0/1/999	0/4/996	0/0/1000	0/4/996	0/1/999	0/3/997	1/4/995	0/3/997

q \ n	15	16	17	18	19	20	21	22
509	1/88/911	11/99/890	6/90/904	3/119/878	3/104/893	7/99/894	6/128/866	3/116/881
1021	1/27/972	3/45/952	5/49/946	1/54/945	5/58/937	2/54/944	2/67/931	7/59/934
2039	4/18/978	1/19/980	0/28/972	2/20/978	2/25/973	2/31/967	2/29/969	2/28/970
4093	2/8/990	1/10/989	1/18/981	0/16/984	3/15/982	1/23/976	1/11/988	1/22/977
8191	1/3/996	0/4/996	1/7/992	0/4/996	1/10/989	1/9/990	0/4/996	0/8/992
16381	0/7/993	0/2/998	0/1/999	0/1/999	0/8/992	0/4/996	0/3/997	1/3/996

Table 7.1: Statistics of effective points. a/b/c in the table are defined as follows: a (resp. b) is the number of points for which Assumption 1 (resp. Assumption 2) does not hold, and c is the number of effective points.

1000 points, and record the number of points failing assumption 1, and the number of points satisfying assumption 1 but failing assumption 2, as well as the number of effective points.

Finally, to verify assumption 3, we do experiments on these effective points. Our results show that for the instances in Table 7.1, the isomorphism invariants corresponding to all points are pairwise distinguishable. This is expected, because each sample is generated randomly, these points are essentially distinct from one another.

Note that it is enough for all but a small fraction of corank-1 $\mathbf{u}_1$ to satisfy the above. Furthermore, if some assumption is not satisfied, this would also constitute as an invariant. That is, if in $L_U, \mathbf{u}_1, \ldots, \mathbf{u}_i$ becomes linearly dependent, then this number i also becomes an invariant which can be utilized. We do not attempt to deal with such cases because they rarely happen in experiments.

7.5 Experimental results for the algorithm

We implemented the algorithm in Section 7.3 in Magma [BJP97]. We tested our implementation on a server (AMD EPYC 7532 CPU at 2.40GHz) to solve some instances

Parameter set	n	q	Number of effective points	Number of sampling times	Time (seconds)
MCE-instance-1	6	61	2702	3721	420
MCE-instance-2	7	61	20053	29062	5638
MCE-instance-3	8	61	149149	226981	100900
MCE-instance-4	9	31	64202	165870	137715

Table 7.2: Solving MCE instances

of the MCE problem. The results are given in Table 7.2. Our experiments demonstrate that when running ten instances, two to four of them successfully discover collisions and recover the secret matrices (A, B, C).

Because we conduct $q^{(n-2)/2}$ samplings, we cannot set q to be too large for a practical running. Therefore, we set q to be 61 or 31. As a result, the fraction of effective points is not as large as for $q = 1021$ as in Table 7.1. For example, in MCE-instance-1, we conducted 3721 samplings and obtained 2702 effective points. Therefore, when q is large, the success rate should increase with the number of effective points.

Remark 7.5.1. Following [Beu23], a possible improvement on the sampling step (Step (a) of the algorithm in Section 7.3) is as follows.

Recall that in Step (a) of the algorithm in Section 7.3, a corank-1 point is obtained by sampling a random vector in $\mathbb{F}_q^n$ for q times. However, note that starting from a corank-1 vector $\hat{\mathbf{u}}$, the vectors in the vector tuple L_U, if successfully built, are all corank-1. So these vectors can be utilized, instead of starting from a fresh random corank-1 vector. In general, we can walk along the path in the tripartite graph starting from a corank-1 vector until we hit a vector of corank larger than 1. This has the potential of reducing the complexity of the algorithm from $O(q^{(n-2)/2} \cdot (q \cdot n^3 + n^4) \cdot (\log(q))^2)$ to $O(q^{(n-2)/2} \cdot n^4 \cdot (\log(q))^2)$, as we would only need to sample a fresh corank-1 vector very few times during the execution of the algorithm.

One question for this approach is whether it results in a distribution close to the uniform one. To test this, we implemented the above approach. In the case of MCE-instance-1, our preliminary experimental results show that when running 6 instances,

one of them successfully finds a collision and recovers the secret matrices. We leave a more careful analysis and more experiments to future work.

Chapter 8

Algorithms for alternating trilinear form equivalence

In this chapter, we present our algorithm for the ATFE problem. That is, given two alternating trilinear forms $\phi \in \mathrm{ATF}(\mathbb{F}_q^n)$ and $\psi \in \mathrm{ATF}(\mathbb{F}_q^n)$, the algorithm computes an equivalence $A \in \mathrm{GL}(n, q)$ from ϕ to ψ, if such A exists.

As will be explained later, there is a component missing for implementing this algorithm for ATFE, namely the transformation of isomorphism testing procedures to canonical forms. (On the contrary, the corresponding component in our algorithm for matrix code equivalence is automatically a canonical form algorithm.) Still, as it is a widely held belief that isomorphism testing procedures can be upgraded to canonical forms, the time complexity of this algorithm is used in the parameter setup of ALTEQ.

Before introducing our algorithm, we review the algorithms for ATFE.

8.1 The direct Gröbner basis attack

Let $\phi, \psi \in \mathrm{ATF}(n, q)$ be two alternating trilinear forms. We wish to decide if there exists $A \in \mathrm{GL}(n, q)$ such that $\phi = \psi \circ A$. The Gröbner basis attack is the following. First, formulate a polynomial system whose solutions are isomorphisms from ϕ to

ψ. Second, use the polynomial solvers, such as Göbner basis and XL, to solve such systems.

Two ways of formulating as polynomial systems are as follows. Both depend on the following data from ϕ and ψ.

From $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ be an alternating trilinear form. Then construct a matrix tuple $\mathrm{A} = (A_1, \ldots, A_n) \in \mathrm{M}(n, q)^n$, where $A_k(i, j) = \phi(e_i, e_j, e_k)$. Recall that e_i is the ith standard basis vector.

Similarly, from $\psi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$, we construct $\mathrm{B} = (B_1, \ldots, B_n) \in \mathrm{M}(n, q)^n$.

The direct cubic modeling. The following modeling is straightforward. Let $X = (x_{i,j})_{i,j\in[n]}$ be an $n \times n$ variable matrices. Set up the following equations.

(1) For $i \in [n]$, set $\sum_{j\in[n]} x_{i,j} \cdot X^t A_i X = B_i$.

Note that by alternating, the above setup uses n^2 variables to set up $\binom{n}{3}$ inhomogeneous equations. Also note that here we do not need to impose that X is invertible, because ϕ and ψ are non-degenerate[1] with high probability.

The quadratic with inverse modelling. This is the formulation studied in [TDJ+22], which traced back to [BFFP11] for cubic form equivalence.

Let $X = (x_{i,j})_{i,j\in[n]}$ and $Y = (y_{i,j})_{i,j\in[n]}$ be two $n \times n$ variable matrices. Set up the following equations.

(1) Set $XY = I_n$ and $YX = I_n$. This imposes that X and Y are inverses to each other.

(2) For $i \in [n]$, set $X^t A_i X = \sum_{j\in[n]} y_{i,j} B_j$, and $Y^t B_i Y = \sum_{j\in[n]} x_{i,j} A_j$.

The above setup uses $2n^2$ variables to set up $2n^2 + 2 \cdot n \cdot \binom{n}{2} = 2n(\binom{n}{2} + n)$ inhomogeneous quadratic equations.

The quadratic dual modeling. This formulation is due to [RST23]. Let $X = (x_{i,j})_{i,j\in[n]}$ be an $n \times n$ variable matrix. Let y be a variable.

[1] ϕ is degenerate if there exists a non-zero vector $u \in \mathbb{F}_q^n$ such that for every $v, w \in \mathbb{F}_q^n$, $\phi(u, v, w) = 0$.

Let $\ell = \binom{n}{2} - n$, and let $C_1, \ldots, C_\ell$ be a basis of the linear space $\{D \in \Lambda(n,q) \mid \mathrm{Tr}(B_i D^t) = 0\}$, where Tr denotes taking the trace of a matrix.

Set up the following equations.

(1) For $i \in [n]$, $j \in [\ell]$, $\mathrm{Tr}(X^t A_i X D^t) = 0$.

(2) Let the $(1,2)$ entry of $X^t A_1 X$ be q, which is a homogeneous quadratic polynomial in $x_{i,j}$. Set $q \cdot y = 1$.

The above setup uses $n^2 + 1$ variables to set up $(\binom{n}{2} - n) \cdot n + 1$ equations. Among them, $(\binom{n}{2} - n) \cdot n$ are homogeneous quadratic polynomials in n^2 variables. The extra cubic equation, $q \cdot y = 1$, is introduced to prevent some undesirable solutions such as rank-1 matrices.

8.2 Beullens' algorithms for ATFE

In [Beu23], Beullens presented some novel algorithms for ATFE. Here we describe two algorithms there that work for general n.

The first algorithm is a collision algorithm based on low-rank points based on the graph-walking sampling method. That is, suppose a random $\phi \in \mathrm{ATF}(n,q)$ has approximately q^k-many projective points of rank r. Then for $\phi, \psi \in \mathrm{ATF}(n,q)$ that are equivalent via $A \in \mathrm{GL}(n,q)$, one can sample $q^{1/2 \cdot k}$-many rank-r points for ϕ, and another $q^{1/2 \cdot k}$-many rank-r points for ψ. Then by the birthday paradox, with constant probability there exists a pair of points $(\mathbf{u}, \mathbf{v})$ from these two lists, such that $A(\mathbf{u}) = \mathbf{v}$. Combined with a Gröbner basis with partial information procedure[2], this correspondence enables to recover the whole A. To sample rank-r points, Beullens invented the graph-walk sampling method, which allows for sampling e.g. corank-3 points for odd n more efficiently than directly using min-rank for relatively small q.

[2]Beullens discovered that Gröbner basis with partial information still works well given that (1) a correspondence between projective points, and (2) the kernel information of low-rank points.

The major cost of this approach is usually the collision step, with time complexity $q^k \cdot \text{poly}(n, \log q)$.

The second algorithm is a birthday algorithm based on isomorphism invariants. Such an algorithm was already proposed for the polynomial isomorphism problem by Bouillaguet, Fouque, and Véber in [BFV13] for $q = 2$. Beullens observed that for radius-1 or 2 neighbors of corank-1 (for odd n) or corank-2 (for even n), the rank information should serve as distinguishing isomorphism invariants. The major cost of this approach is the number of corank-1 or corank-2 points, so Beullens estimated the running time as $q^{n/2+c} \cdot \text{poly}(n, \log q)$.

8.3 An algorithm for ATFE based on a new isomorphism invariant

The main innovation of our algorithm for ATFE is to associate distinguishing isomorphism invariants to low-rank points.

Let $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$. Suppose by Theorem 6.1.3, it is expected that there are roughly q^k many $\hat{\mathbf{u}} \in \mathbb{P}(\mathbb{F}_q^n)$, such that $\text{rk}_\phi(\hat{\mathbf{u}}) = r$. Let us *assume* that there is an easy-to-compute, distinguishing, isomorphism invariant[3] for those rank-r $\hat{\mathbf{u}}$.

Then the algorithm goes as follows: first sample $O(q^{k/2})$-many rank-r points for ϕ, and $O(q^{k/2})$-many rank-r points for ψ. For each point, compute this isomorphism invariant. Then by the birthday paradox, there exists one point $\hat{\mathbf{u}}$ from the list of ϕ, and one point $\hat{\mathbf{v}}$ from the list of ψ, such that their isomorphism invariants are the same. Finally, use Gröbner basis with partial information for $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ to recover the desired isomorphism.

Following Equation 6.1, the running time of the above algorithm can then be estimated as

$$O(q^{k/2} \cdot (\text{samp-cost} + \text{inv-cost}) + \text{gb-cost}),$$

[3]That is a function f from low-rank points to some set S, such that $f(\hat{\mathbf{u}}) \neq f(\hat{\mathbf{v}})$ for $\hat{\mathbf{u}} \neq \hat{\mathbf{v}}$, and f is unchanged by basis changes.

where samp-cost denotes the sampling cost, the inv-cost denotes the invariant computing cost, and gb-cost denotes the Gröbner basis with partial information cost.

The sampling step can be achieved by either the min-rank method (Section 8.4) or Beullens' graph-walking method [Beu23]. For the min-rank method, it can be estimated for concrete values of n, k, and r by e.g. [BBC+20, KS99, VBC+19]. For the graph-walking method, it can be estimated based on certain statistics of graphs associated with alternating trilinear forms by Beullens [Beu23, Theorem 1].

The gb-cost can be estimated as $O(n^6)$ as in [Beu23]. This is based on the hybrid Gröbner basis method with the first row known in the variable matrix. The effectiveness of this hybrid Gröbner basis method was first discovered in [FP06] and then utilized in [BFV13, TDJ+22]. Beullens further improved this method by noting that (1) knowing the first row up to scalar suffices, and (2) for low-rank points, the kernel information can be incorporated [Beu23, Section 4].

The main innovation of the above algorithm is a new isomorphism invariant which we describe next.

8.4 Low-rank point sampling via min-rank step

The sampling step can be done by either the min-rank method, or the graph-walking method. The graph-walking method involves q, so it works best for relatively small q. When q is large, the min-rank method is more effective. To use min-rank to do sampling requires a bit of twist, so we record the idea here.

Suppose we wish to sample a rank-r point $\hat{\mathbf{v}} \in \mathbb{P}(\mathbb{F}_q^n)$ for an alternating trilinear form ϕ, and suppose that there are q^k-many rank-r projective points for a random ϕ. To sample such points, we make a heuristic assumption that the first k coordinates of these rank-r points are in uniform random. Therefore, to sample one point, we can randomly choose the first k coordinates and then resort to the min-rank procedure.

More specifically, for $i \in [n]$, let A_i be the alternating matrix representing the bilinear form ϕ_{e_i}, where e_i is the ith standard basis vector. Let x_i, $i \in [n]$, be formal

variables, and set $A = \sum_{i\in[n]} x_i A_i$. So for $i \in [1 \dots k]$, let $x_i = \alpha_i x_1$, where $\alpha_i \in_R \mathbb{F}_q$. This gives us a min-rank instance with $n - k$ matrices of size $n \times n$.

To estimate the min-rank cost, we use the algorithm from [BBC+20]. Consider an (n, K, r) minrank instance, namely finding a rank-r matrix in a linear span of K $n \times n$ matrices. First, we need to compute the smallest b such that $b < r + 2$ and

$$\binom{n}{r}\binom{K+b-1}{b} - 1 \leq \sum_{i=1}^{b} (-1)^{i+1} \binom{n}{r+i}\binom{n+i-1}{i}\binom{K+b-i-1}{b-i}.$$

Based on this b, the complexity is estimated as

$$O\big(K \cdot (r+1) \cdot (\binom{n}{r} \cdot \binom{K+b-1}{b})^2\big).$$

For concrete values of n, $K = n - k$ and r, the above formulas allow for the estimation of the concrete security parameters.

Note that the min-rank instance above has some structural constraints due to alternating trilinear forms. As pointed out in [Beu23], such structures should impact the min-rank algorithm from [BBC+20] adversely. Still, we use the estimates from [BBC+20] as they should serve as a lower bound. We also compare the estimates from [BBC+20] with the analysis of the Kipnis–Shamir modeling [KS99] in [VBC+19], and found the ones from [BBC+20] are lower.

8.5 The isomorphism invariant step

Suppose $\hat{\mathbf{u}} \in \mathbb{P}(\mathbb{F}_q^n)$ satisfies that $\mathrm{rk}_\phi(\hat{\mathbf{u}}) = r$. Then $K := \ker(\phi_{\hat{\mathbf{u}}}) \leq \mathbb{F}_q^n$ is a dimension-$(n - r)$ space, also preserved by any isomorphism. This allows us to consider the trilinear form $\tilde{\phi}_{\hat{\mathbf{u}}} : K \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$, and it can be verified easily that the *isomorphism type* of $\tilde{\phi}_{\hat{\mathbf{u}}}$ under $\mathrm{GL}(K) \times \mathrm{GL}(n, q)$ is an isomorphism invariant.

To use the isomorphism type of $\tilde{\phi}_{\hat{\mathbf{u}}}$ in the algorithm, we need the isomorphism types to be (1) easy to compute, and (2) distinguishing; that is, for different $\hat{\mathbf{u}}, \hat{\mathbf{v}} \in \mathbb{P}(\mathbb{F}_q^n)$, $\tilde{\phi}_{\hat{\mathbf{u}}}$ and $\tilde{\phi}_{\hat{\mathbf{v}}}$ are different.

To verify these, we perform the following experiment in Magma [BJP97].

(1) Sample a random $\phi \in \mathrm{ATF}(n, q)$.

(2) Sample a random rank-r point $\hat{\mathbf{u}} \in \mathbb{P}(\mathbb{F}_q^n)$.

(3) Sample t random rank-r point $\hat{\mathbf{v}} \in \mathbb{P}(\mathbb{F}_q^n)$. For each such point, do:

a) Use the Gröbner basis with partial information to decide whether $\tilde{\phi}_{\hat{\mathbf{u}}}$ and $\tilde{\phi}_{\hat{\mathbf{v}}}$ are isomorphic.

Our experiments give the following.

- For $n = 9$, $r = 4$, and $p = 3$, 10 experiments (i.e. for 10 $\hat{\mathbf{u}}$ from 10 random alternating trilinear forms) with $t = 100$ comparisons (i.e. for 100 different $\hat{\mathbf{v}}$ to compare with $\hat{\mathbf{u}}$). On average, 75 out of 100 $\tilde{\phi}_{\hat{\mathbf{v}}}$ are not isomorphic with $\tilde{\phi}_{\hat{\mathbf{u}}}$.
- For $n = 10$, $r = 6$, and $p = 3$, 10 experiments (i.e. for 10 $\hat{\mathbf{u}}$ from 10 random alternating trilinear forms) with $t = 100$ comparisons (i.e. for 100 different $\hat{\mathbf{v}}$ to compare with $\hat{\mathbf{u}}$) all return "different isomorphism type". On average, 95 out of 100 $\tilde{\phi}_{\hat{\mathbf{v}}}$ are not isomorphic with $\tilde{\phi}_{\hat{\mathbf{u}}}$.

For $n = 11$, our code do not work for $n = 11$ on a laptop, due to the Gröbner basis step.

From these experiments we see that (1) the Gröbner basis with partial information algorithm is effective in practice to test isomorphism between $\tilde{\phi}_{\hat{\mathbf{u}}}$ and $\tilde{\phi}_{\hat{\mathbf{v}}}$, and (2) the isomorphism type of $\tilde{\phi}_{\hat{\mathbf{u}}}$ is close to distinguishing for $n = 10$. These give support that the isomorphism types of $\tilde{\phi}_{\hat{\mathbf{u}}}$ do serve as a easy-to-compute, distinguishing, isomorphism invariant.

Note that testing isomorphism here is not enough, and canonical forms are required to serve as an isomorphism invariant. Even though to transform an isomorphism invariant algorithm to a canonical form one may not be an easy process, it is generally regarded as doable, at least from the experience from graph isomorphism [Bab16]. We leave the construction of the canonical form as future work.

Bibliography

[ABB+22] Nicolas Aragon, Paulo Barreto, Slim Bettaieb, Loic Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Phillipe Gaborit, Shay Gueron, Tim Guneysu, Carlos Aguilar Melchor, Rafael Misoczki, Edoardo Persichetti, Nicolas Sendrier, Jean-Pierre Tillich, Gilles Zemor, Valentin Vasseur, Santosh Ghosh, and Jan Richter-Brokmann. Bike: Bit flipping key encapsulation. https://bikesuite.org/, 2022.

[ABWB+20] Jean-Philippe Aumasson, Daniel J. Bernstein, Christoph Dobraunig Ward Beullens, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Andreas Hülsing, Panos Kampanakis, Stefan KÏbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, and Bas Westerbaan. Sphincs+: Submission to the nist post-quantum project, v.3. https://sphincs.org/data/sphincs+-round3-specification.pdf, 2020.

[AFLT12] Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-secure signatures from lossy identification schemes. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer*

Science, pages 572–590. Springer, 2012. doi:10.1007/978-3-642-29011-4_34.

[AFMP20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 411–439. Springer, 2020. doi:10.1007/978-3-030-64834-3_14.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 99–108. ACM, 1996. doi:10.1145/237814.237838.

[AS05] Manindra Agrawal and Nitin Saxena. Automorphisms of finite rings and applications to complexity of problems. In *STACS 2005, 22nd Annual Symposium on Theoretical Aspects of Computer Science, Stuttgart, Germany, February 24-26, 2005, Proceedings*, pages 1–17, 2005. doi:10.1007/978-3-540-31856-9_1.

[AS06] Manindra Agrawal and Nitin Saxena. Equivalence of $\mathbb{F}$-algebras and cubic forms. In *STACS 2006, 23rd Annual Symposium on Theoretical Aspects of Computer Science, Proceedings*, pages 115–126, 2006. doi:10.1007/11672142_8.

[Atk73] MD Atkinson. Alternating trilinear forms and groups of exponent 6. *Journal of the Australian Mathematical Society*, 16(1):111–128, 1973.

[Bab16] László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 684–697, 2016.

[BBB+23] Marco Baldi, Alessandro Barenghi, Luke Beckwith, Jean-François Biasse, Andre Esser, Kris Gaj, Kamyar Mohajerani, Gerardo Pelosi, Edoardo Persichetti, Markku-Juhani Saarinen, Paolo Santini, and Robert Wallace. Less: Linear equivalence signature scheme, 2023. URL https://www.less-project.com/LESS-2023-08-18.pdf.

[BBC+20] Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray A. Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier A. Verbel. Improvements of algebraic attacks for solving the rank decoding and minrank problems. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 507–536. Springer, 2020. doi:10.1007/978-3-030-64837-4_17.

[BBMP23] Michele Battagliola, Giacomo Borin, Alessio Meneghetti, and Edoardo Persichetti. Cutting the grass: Threshold group action signature schemes. *Cryptology ePrint Archive*, 2023.

[BBN+22] Alessandro Barenghi, Jean-François Biasse, Tran Ngo, Edoardo Persichetti, and Paolo Santini. Advanced signature functionalities from the code equivalence problem. *International Journal of Computer Mathematics: Computer Systems Theory*, 7(2):112–128, 2022.

[BBPS21] Alessandro Barenghi, Jean-François Biasse, Edoardo Persichetti, and Paolo Santini. Less-fm: fine-tuning signatures from the code equivalence

problem. In *Post-Quantum Cryptography: 12th International Workshop, PQCrypto 2021, Daejeon, South Korea, July 20–22, 2021, Proceedings 12*, pages 23–43. Springer, 2021.

[BCC+22] Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Edoardo Persichetti, Christiane Peters, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, and Wen Wang. Classic mceliece. https://classic.mceliece.org/, 2022.

[BCD+22] Markus Bläser, Zhili Chen, Dung Hoang Duong, Antoine Joux, Ngoc Tuong Nguyen, Thomas Plantard, Youming Qiao, Willy Susilo, and Gang Tang. On digital signatures based on isomorphism problems: Qrom security, ring signatures, and applications. Cryptology ePrint Archive, Paper 2022/1184, 2022. URL https://eprint.iacr.org/2022/1184.

[BCD+23] Ward Beullens, Ming-Shing Chen, Jintai Ding, Boru Gong, Matthias J. Kannwischer, Jacques Patarin, Bo-Yuan Peng, Dieter Schmidt, Cheng-Jhih Shih, Chengdong Tao, and Bo-Yin Yang. Uov: Unbalanced oil and vinegar. https://www.uovsig.org/, 2023.

[BCH+23] Ward Beullens, Ming-Shing Chen, Shih-Hao Hung, Matthias J. Kannwischer, Bo-Yuan Peng, Cheng-Jhih Shih, and Bo-Yin Yang. Oil and vinegar: Modern parameters and implementations. *IACR Cryptol. ePrint Arch.*, page 59, 2023. URL https://eprint.iacr.org/2023/059.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, De-*

cember 4-8, 2011. Proceedings, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011. doi:10.1007/978-3-642-25385-0_3.

[BDK+21] Shi Bai, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: Algorithm specifications and supporting documentation (version 3.1). https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf, 2021.

[BDK+22] Ward Beullens, Samuel Dobson, Shuichi Katsumata, Yi-Fu Lai, and Federico Pintore. Group signatures and more from isogenies and lattices: Generic, simple, and efficient. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part II*, volume 13276 of *Lecture Notes in Computer Science*, pages 95–126. Springer, 2022. doi:10.1007/978-3-031-07085-3_4.

[BDN+23] Markus Bläser, Dung Hoang Duong, Anand Kumar Narayanan, Thomas Plantard, Youming Qiao, Arnaud Sipasseuth, and Gang Tang. The alteq signature scheme: Algorithm specifications and supporting documentation, 2023. URL https://pqcalteq.github.io/ALTEQ_spec_2023.09.18.pdf.

[Bel93] E Belsley. *Rates of convergence of Markov chains related to association schemes, Harvard University Ph. D.* PhD thesis, thesis, 1993.

[BESV22] Emanuele Bellini, Andre Esser, Carlo Sanna, and Javier Verbel. Mr-dss–smaller minrank-based (ring-) signatures. In *Post-Quantum Cryptography: 13th International Workshop, PQCrypto 2022, Virtual Event, September 28–30, 2022, Proceedings*, pages 144–169. Springer, 2022.

[Beu20] Ward Beullens. Not enough less: An improved algorithm for solving code equivalence problems over f q. In *International Conference on Selected Areas in Cryptography*, pages 387–403. Springer, 2020.

[Beu23] Ward Beullens. Graph-theoretic algorithms for the alternating trilinear form equivalence problem. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 101–126. Springer, 2023. doi:10.1007/978-3-031-38548-3_4.

[BFFP11] Charles Bouillaguet, Jean-Charles Faugère, Pierre-Alain Fouque, and Ludovic Perret. Practical cryptanalysis of the identification scheme based on the isomorphism of polynomial with one secret problem. In *International Workshop on Public Key Cryptography*, pages 473–493. Springer, 2011. doi:10.1007/978-3-642-19379-8_29.

[BFGP23] Ward Beullens, Luca De Feo, Steven D. Galbraith, and Christophe Petit. Proving knowledge of isogenies: a survey. *Des. Codes Cryptogr.*, 91(11):3425–3456, 2023. doi:10.1007/S10623-023-01243-3.

[BFP15] Jérémy Berthomieu, Jean-Charles Faugère, and Ludovic Perret. Polynomial-time algorithms for quadratic isomorphism of polynomials: The regular case. *J. Complexity*, 31(4):590–616, 2015. doi:10.1016/j.jco.2015.04.001.

[BFSY05] Magali Bardet, Jean-Charles Faugère, Bruno Salvy, and Bo-Yin Yang. Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. In *Proc. of MEGA*, volume 5, 2005.

[BFV13] Charles Bouillaguet, Pierre-Alain Fouque, and Amandine Véber. Graph-theoretic algorithms for the "isomorphism of polynomials" problem. In *Advances in Cryptology - EUROCRYPT 2013*, pages 211–227, 2013.

[BJP97] W. Bosma, J. J. Cannon, and C. Playoust. The Magma algebra system I: the user language. *J. Symb. Comput.*, pages 235–265, 1997.

[BKP20] Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 464–492. Springer, 2020. doi:10.1007/978-3-030-64834-3_16.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I*, volume 11921 of *Lecture Notes in Computer Science*, pages 227–247. Springer, 2019. doi:10.1007/978-3-030-34578-5_9.

[BLQW20] Peter A. Brooksbank, Yinan Li, Youming Qiao, and James B. Wilson. Improved algorithms for alternating matrix space isometry: From theory to practice. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 26:1–26:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ESA.2020.26.

[BMPS20] Jean-François Biasse, Giacomo Micheli, Edoardo Persichetti, and Paolo Santini. LESS is more: Code-based signatures without syndromes. In Abderrahmane Nitaj and Amr M. Youssef, editors, *Progress in Cryptology - AFRICACRYPT 2020 - 12th International Conference on Cryptology in Africa, Cairo, Egypt, July 20-22, 2020, Proceedings*, volume 12174 of *Lecture Notes in Computer Science*, pages 45–65. Springer, 2020. doi:10.1007/978-3-030-51938-4_3.

[BMVT78] Elwyn Berlekamp, Robert McEliece, and Henk Van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[BMW17] Peter A. Brooksbank, Joshua Maglione, and James B. Wilson. A fast isomorphism test for groups whose Lie algebra has genus 2. *J. Algebra*, 473:545–590, 2017.

[BN06] Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, October 30 - November 3, 2006*, pages 390–399. ACM, 2006. doi:10.1145/1180405.1180453.

[Bon98] Dan Boneh. The decision Diffie-Hellman problem. In *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, pages 48–63, 1998. doi:10.1007/BFb0054851.

[BOST19] Magali Bardet, Ayoub Otmani, and Mohamed Saeed-Taha. Permutation code equivalence is not harder than graph isomorphism when hulls are trivial. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 2464–2468. IEEE, 2019. doi:10.1109/ISIT.2019.8849855.

[Bou11] Charles Bouillaguet. *Etudes d'hypothèses algorithmiques et attaques de primitives cryptographiques*. PhD thesis, PhD thesis, Université Paris-Diderot–École Normale Supérieure, 2011.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[BS20] Xavier Bonnetain and André Schrottenloher. Quantum security analysis of CSIDH. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 493–522. Springer, 2020. doi:10.1007/978-3-030-45724-2_17.

[BVWW16] Zvika Brakerski, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Obfuscating conjunctions under entropic ring LWE. In Madhu Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 147–156. ACM, 2016. doi:10.1145/2840728.2840764.

[BY90] Gilles Brassard and Moti Yung. One-way group actions. In *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, pages 94–107, 1990. doi:10.1007/3-540-38424-3_7.

[CD23] Wouter Castryck and Thomas Decru. An efficient key recovery attack on sidh. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 423–447. Springer, 2023.

[CDAG20] Alain Couvreur, Thomas Debris-Alazard, and Philippe Gaborit. On the hardness of code equivalence problems in rank metric. arXiv preprint arXiv:2011.04611, 2020.

[CGQ+24] Zhili Chen, Joshua A. Grochow, Youming Qiao, Gang Tang, and Chuanqi Zhang. On the complexity of isomorphism problems for tensors, groups, and polynomials III: actions by classical groups. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, January 30 to February 2, 2024, Berkeley, CA, USA*, volume 287 of *LIPIcs*, pages 31:1–31:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ITCS.2024.31.

[CH88] Arjeh M. Cohen and Aloysiu G. Helminck. Trilinear alternating forms on a vector space of dimension 7. *Communications in algebra*, 16(1):1–25, 1988.

[CJS14] Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014. doi:10.1515/jmc-2012-0016.

[CLG09] Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *J. Cryptol.*, 22(1):93–113, 2009. doi:10.1007/S00145-007-9002-X.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)*, pages 395–427. Springer, 2018.

[CNP+23a] Tung Chou, Ruben Niederhagen, Edoardo Persichetti, Lars Ran, Tovohery Hajatiana Randrianarisoa, Krijn Reijnders, Simona Samard-

jiska, and Monika Trimoska. Matrix code equivalence digital signature, 2023. URL https://www.meds-pqc.org/spec/MEDS-2023-07-26.pdf.

[CNP+23b] Tung Chou, Ruben Niederhagen, Edoardo Persichetti, Tovohery Hajatiana Randrianarisoa, Krijn Reijnders, Simona Samardjiska, and Monika Trimoska. Take your MEDS: digital signatures from matrix code equivalence. In Nadia El Mrabet, Luca De Feo, and Sylvain Duquesne, editors, *Progress in Cryptology - AFRICACRYPT 2023 - 14th International Conference on Cryptology in Africa, Sousse, Tunisia, July 19-21, 2023, Proceedings*, volume 14064 of *Lecture Notes in Computer Science*, pages 28–52. Springer, 2023. doi:10.1007/978-3-031-37679-5_2.

[Cou06] Jean Marc Couveignes. Hard homogeneous spaces. *IACR Cryptology ePrint Archive*, 2006. URL http://eprint.iacr.org/2006/291.

[Cra92] R. E. Crandall. Method and apparatus for public key exchange in a cryptographic system, 1992. *U.S. Patent number 5159632.*

[CSSF+23] Jorge Chavez-Saab, Maria Corte-Real Santos, Luca De Feo, Jonathan Komada Eriksen, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Michael Meyer, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Francisco Rodríguez Henríquez, Sina Schaeffler, and Benjamin Wesolowski. Sqisign: compact post-quantum signatures from quaternions and isogenies. https://sqisign.org/, 2023.

[DFKL+20] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. Sqisign: compact post-quantum signatures from quaternions and isogenies. In International Conference on the Theory and Application of Cryptology and Information Security, *pages 64–93. Springer, 2020.*

[DFMS19] *Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the fiat-shamir transformation in the quantum random-oracle model. In Alexandra Boldyreva and Daniele Micciancio, editors,* Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II, *volume 11693 of* Lecture Notes in Computer Science, *pages 356–383. Springer, 2019. doi:10.1007/978-3-030-26951-7_13.*

[DG22] *Giuseppe D'Alconzo and Andrea Gangemi. Trifors: Linkable trilinear forms ring signature.* Cryptology ePrint Archive, *2022.*

[DH76] *Whitfield Diffie and Martin Hellman. New directions in cryptography.* IEEE Transactions on Information Theory, *22(6):644–654, 1976.*

[DN19] *Itai Dinur and Niv Nadler. Multi-target attacks on the picnic signature scheme and related protocols. In Yuval Ishai and Vincent Rijmen, editors,* Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III, *volume 11478 of* Lecture Notes in Computer Science, *pages 699–727. Springer, 2019. doi:10.1007/978-3-030-17659-4_24.*

[DPPW22] *Léo Ducas, Eamonn W Postlethwaite, Ludo N Pulles, and Wessel van Woerden. Hawk: Module lip makes lattice signatures fast, compact and simple. In* International Conference on the Theory and Application of Cryptology and Information Security, *pages 65–94. Springer, 2022.*

[DS05] *Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In* International Conference on Applied Cryptography and Network Security, *pages 164–175. Springer, 2005.*

[DS14] *Jan Draisma and Ron Shaw. Some noteworthy alternating trilinear forms.* Journal of Geometry, *105(1):167–176, 2014.*

[DvW22] *Léo Ducas and Wessel van Woerden. On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In* Advances in Cryptology–EUROCRYPT 2022: 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30–June 3, 2022, Proceedings, Part III*, pages 643–673. Springer, 2022.*

[EKP20] *Ali El Kaafarani, Shuichi Katsumata, and Federico Pintore. Lossy csi-fish: Efficient signature scheme with tight reduction to decisional CSIDH-512. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors,* Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part II*, volume 12111 of* Lecture Notes in Computer Science*, pages 157–186. Springer, 2020. doi:10.1007/978-3-030-45388-6_6.*

[ESZ22] *Muhammed F Esgin, Ron Steinfeld, and Raymond K Zhao. Matrict+: More efficient post-quantum private blockchain payments. In* 2022 IEEE Symposium on Security and Privacy (SP)*, pages 1281–1298. IEEE, 2022.*

[EZS+19] *Muhammed F Esgin, Raymond K Zhao, Ron Steinfeld, Joseph K Liu, and Dongxi Liu. Matrict: efficient, scalable and post-quantum blockchain confidential transactions protocol. In* Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 567–584, 2019.*

[FG15] *Jason Fulman and Larry Goldstein. Stein's method and the rank distribution of random matrices over finite fields.* The Annals of Probability*, 43(3), may 2015. doi:10.1214/13-aop889.*

[FG19] *Luca De Feo and Steven D. Galbraith. Seasign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors,* Advances in Cryptology – EUROCRYPT 2019*, volume 11478 of* Lecture Notes in Computer Science*, pages 759–789. Springer, 2019. doi:10.1007/978-3-030-17659-4_26.*

[FGH+12] *Edward Farhi, David Gosset, Avinatan Hassidim, Andrew Lutomirski, and Peter Shor. Quantum money from knots. In* Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 276–289, 2012.*

[FGS19] *Vyacheslav Futorny, Joshua A. Grochow, and Vladimir V. Sergeichuk. Wildness for tensors.* Linear Algebra and its Applications*, 566:212–244, 2019. doi:10.1016/j.laa.2018.12.022.*

[FHK+20] *Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-fourier lattice-based compact signatures over ntru (specification v1.2).* https://falcon-sign.info/falcon.pdf*, 2020.*

[FM20] *Luca De Feo and Michael Meyer. Threshold schemes from isogeny assumptions. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors,* Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part II*, volume 12111 of* Lecture Notes in Computer Science*, pages 187–212. Springer, 2020. doi:10.1007/978-3-030-45388-6_7.*

[FP06] *Jean-Charles Faugère and Ludovic Perret. Polynomial equivalence problems: Algorithmic and theoretical aspects. In* Advances in Cryptology - EUROCRYPT 2006*, pages 30–47, 2006.*

[FS86] *Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In* Advances in Cryptology – CRYPTO 1986, *pages 186–194, 1986.*

[Gen09] *Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor,* Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009, *pages 169–178. ACM, 2009. doi:10.1145/1536414.1536440.*

[GMW91] *Oded Goldreich, Silvio Micali, and Avi 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. doi:10.1145/116825.116852.*

[GQ21a] *Joshua A. Grochow and Youming Qiao. On p-group isomorphism: Search-to-decision, counting-to-decision, and nilpotency class reductions via tensors. In Valentine Kabanets, editor,* 36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference), *volume 200 of* LIPIcs, *pages 16:1–16:38. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.CCC.2021.16.*

[GQ21b] *Joshua A. Grochow and Youming Qiao. On the complexity of isomorphism problems for tensors, groups, and polynomials I: tensor isomorphism-completeness. In James R. Lee, editor,* 12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference, *volume 185 of* LIPIcs, *pages 31:1–31:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.31.*

[GQT21] *Joshua A. Grochow, Youming Qiao, and Gang Tang. Average-case algorithms for testing isomorphism of polynomials, algebras, and multilinear forms. In Markus Bläser and Benjamin Monmege, editors,* 38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference),

volume 187 of LIPIcs*, pages 38:1–38:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.STACS.2021.38.*

[GSVV04] *Michelangelo Grigni, Leonard J. Schulman, Monica Vazirani, and Umesh V. Vazirani. Quantum mechanical algorithms for the nonabelian hidden subgroup problem.* Comb.*, 24(1):137–154, 2004. doi:10.1007/s00493-004-0009-8.*

[GVW13] *Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors,* Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 545–554. ACM, 2013. doi:10.1145/2488608.2488677.*

[GVW15] *Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew Robshaw, editors,* Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of* Lecture Notes in Computer Science*, pages 503–523. Springer, 2015. doi:10.1007/978-3-662-48000-7_25.*

[Hås90] *Johan Håstad. Tensor rank is NP-complete.* J. Algorithms*, 11(4):644–654, 1990. doi:10.1016/0196-6774(90)90014-6.*

[HMR+10] *Sean Hallgren, Cristopher Moore, Martin Rötteler, Alexander Russell, and Pranab Sen. Limitations of quantum coset states for graph isomorphism.* J. ACM*, 57(6):34:1–34:33, November 2010. doi:10.1145/1857914.1857918.*

[HP15] *Jan Hora and Petr Pudlák. Classification of 8-dimensional trilinear alternating forms over gf (2).* Communications in Algebra*, 43(8):3459–3471, 2015.*

[HP21] *Jan Hora and Petr Pudlák. Classification of 9-dimensional trilinear alternating forms over gf (2).* Finite Fields and Their Applications, *70:101788, 2021.*

[IQ19] *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. doi:10.1137/18M1165682.*

[JAC+22] *David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, David Urbanik, Geovandro Pereira, Koray Karabina, and Aaron Hutchinson. Sike: Supersingular isogeny key encapsulation.* https://sike.org/, *2022.*

[JF11] *David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor,* Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings, *volume 7071 of* Lecture Notes in Computer Science, *pages 19–34. Springer, 2011. doi:10.1007/978-3-642-25405-5_2.*

[JQSY19] *Zhengfeng Ji, Youming Qiao, Fang Song, and Aaram Yun. General linear group action on tensors: A candidate for post-quantum cryptography. In Dennis Hofheinz and Alon Rosen, editors,* Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part I, *volume 11891 of* Lecture Notes in Computer Science, *pages 251–281. Springer, 2019. doi:10.1007/978-3-030-36030-6_11.*

[KLS18] *Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In*

Advances in Cryptology – EUROCRYPT 2018, *pages 552–586. Springer, 2018.*

[KPG99] *Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. In Jacques Stern, editor,* Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding, *volume 1592 of* Lecture Notes in Computer Science, *pages 206–222. Springer, 1999. doi:10.1007/3-540-48910-X_15.*

[KS98] *Aviad Kipnis and Adi Shamir. Cryptanalysis of the oil & vinegar signature scheme. In Hugo Krawczyk, editor,* Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings, *volume 1462 of* Lecture Notes in Computer Science, *pages 257–266. Springer, 1998. doi:10.1007/BFB0055733.*

[KS99] *Aviad Kipnis and Adi Shamir. Cryptanalysis of the hfe public key cryptosystem by relinearization. In* Annual International Cryptology Conference, *pages 19–30. Springer, 1999.*

[KST93] *Johannes Köbler, Uwe Schöning, and Jacobo Torán.* The graph isomorphism problem: its structural complexity. *Birkhauser Verlag, Basel, Switzerland, Switzerland, 1993. doi:10.1007/978-1-4612-0333-9.*

[Lam79] *Leslie Lamport. Constructing digital signatures from a one-way function. Technical report, Technical Report CSL-98, SRI International, 1979.*

[Lan02] *Serge Lang.* Algebra. *Number 211 in Graduate Texts in Mathematics. Springer-Verlag, New York, third enlarged edition, 2002.*

[LAZ19] *Xingye Lu, Man Ho Au, and Zhenfei Zhang. Raptor: A practical lattice-based (linkable) ring signature. In Robert H. Deng, Valérie Gauthier-*

Umaña, Martín Ochoa, and Moti Yung, editors, Applied Cryptography and Network Security*, pages 110–130, Cham, 2019. Springer International Publishing.*

[Leo82] *Jeffrey Leon. Computing automorphism groups of error-correcting codes.* IEEE Transactions on Information Theory, *28(3):496–511, 1982.*

[LNS21] *Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Smile: set membership from ideal lattices with applications to ring signatures and confidential transactions. In* Annual International Cryptology Conference*, pages 611–640. Springer, 2021.*

[LQ17] *Yinan Li and Youming Qiao. Linear algebraic analogues of the graph isomorphism problem and the Erdős–Rényi model. In Chris Umans, editor,* 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 463–474. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.49. arXiv:1708.04501, version 2.*

[LW05] *Joseph K. Liu and Duncan S. Wong. Linkable ring signatures: Security models and new schemes. In Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganà, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan, editors,* Computational Science and Its Applications - ICCSA 2005, International Conference, Singapore, May 9-12, 2005, Proceedings, Part II*, volume 3481 of* Lecture Notes in Computer Science*, pages 614–623. Springer, 2005. doi:10.1007/11424826_65.*

[LZ19] *Qipeng Liu and Mark Zhandry. Revisiting post-quantum fiat-shamir. In Alexandra Boldyreva and Daniele Micciancio, editors,* Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, volume 11693 of* Lecture Notes in Computer Science*, pages 326–355. Springer, 2019. doi:10.1007/978-3-030-26951-7_12.*

[MAB+22] Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, Gilles Zémor, Jurjen Bos, Arnaud Dion, Jerome Lacan, Jean-Marc Robert, and Pascal Veron. Hqc: Hamming quasi-cyclic. https://pqc-hqc.org/, 2022.

[McE78] R. J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. Deep Space Network Progress Report, 44:114–116, Jan 1978.

[McK80] Brendan D. McKay. Practical graph isomorphism. Congr. Numer., pages 45–87, 1980.

[Mer89] Ralph C Merkle. A certified digital signature. In Conference on the Theory and Application of Cryptology, pages 218–238. Springer, 1989.

[MI88] Tsutomu Matsumoto and Hideki Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In Advances in Cryptology—EUROCRYPT'88: Workshop on the Theory and Application of Cryptographic Techniques Davos, Switzerland, May 25–27, 1988 Proceedings 7, pages 419–453. Springer, 1988.

[MMP+23] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on sidh. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 448–471. Springer, 2023.

[MN13] 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.

[MP14] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. J. Symb. Comput., 60:94–112, 2014.

[MRS08] *Cristopher Moore, Alexander Russell, and Leonard J. Schulman. The symmetric group defies strong fourier sampling.* SIAM J. Comput., *37(6):1842–1864, 2008. doi:10.1137/050644896.*

[MRV07] *Cristopher Moore, Alexander Russell, and Umesh Vazirani. A classical one-way function to confound quantum adversaries.* arXiv preprint quant-ph/0701115, *2007.*

[O'B94] *Eamonn A O'Brien. Isomorphism testing for p-groups.* Journal of Symbolic Computation, *17(2):133–147, 1994.*

[oST22] *National Institute of Standards and Technology. Call for additional digital signature schemes for the post-quantum cryptography standardization process, October 2022. URL* https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf.

[Pat95] *Jacques Patarin. Cryptanalysis of the matsumoto and imai public key scheme of eurocrypt'88. In Don Coppersmith, editor,* Advances in Cryptology - CRYPTO '95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 27-31, 1995, Proceedings, *volume 963 of* Lecture Notes in Computer Science, *pages 248–261. Springer, 1995. doi:10.1007/3-540-44750-4_20.*

[Pat96] *Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In* Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding, *pages 33–48, 1996. doi:10.1007/3-540-68339-9_4.*

[Pei20] *Chris Peikert. He gives c-sieves on the CSIDH. In Anne Canteaut and Yuval Ishai, editors,* Advances in Cryptology - EUROCRYPT 2020 - 39th An-

nual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II, *volume 12106 of* Lecture Notes in Computer Science, *pages 463–492. Springer, 2020.* doi:10.1007/978-3-030-45724-2_16.

[Reg04] *Oded Regev. Quantum computation and lattice problems.* SIAM J. Comput., *33(3):738–760, 2004.* doi:10.1137/S0097539703440678.

[Reg05] *Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors,* Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005, *pages 84–93. ACM, 2005.* doi:10.1145/1060590.1060603.

[Rob23] *Damien Robert. Breaking sidh in polynomial time. In* Annual International Conference on the Theory and Applications of Cryptographic Techniques, *pages 472–503. Springer, 2023.*

[RS06] *Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies.* IACR Cryptol. ePrint Arch., *page 145, 2006. URL* http://eprint.iacr.org/2006/145.

[RST01] *Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor,* Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings, *volume 2248 of* Lecture Notes in Computer Science, *pages 552–565. Springer, 2001.* doi:10.1007/3-540-45682-1_32.

[RST23] *Lars Ran, Simona Samardjiska, and Monika Trimoska. Algebraic algorithm for the alternating trilinear form equivalence problem. In* Code-Based Cryptography Workshop, *pages 84–103. Springer, 2023.*

[RST24] *Krijn Reijnders, Simona Samardjiska, and Monika Trimoska. Hardness estimates of the code equivalence problem in the rank metric.* Designs, Codes and Cryptography, *pages 1–30, 2024.*

[Saa23] *Markku-Juhani O. Saarinen. A forgery attack against alteq, 2023. URL* https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/-LCPCJCyL1c/m/_ghV61NQBQAJ?pli=1.

[SAB+22] *Peter Schwabe, Roberto Avanzi, Joppe Bos, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, Damien Stehle, and Jintai Ding. Crystals-kyber.* https://pq-crystals.org/kyber/index.shtml, *2022.*

[SALY17] *Shifeng Sun, Man Ho Au, Joseph K. Liu, and Tsz Hon Yuen. Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors,* Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II, *volume 10493 of* Lecture Notes in Computer Science, *pages 456–474. Springer, 2017. doi:10.1007/978-3-319-66399-9_25.*

[sCDK+21] *Ming shing Chen, Jintai Ding, Matthias Kannwischer, Jacques Patarin, Albrecht Petzoldt, Dieter Schmidt, and Bo-Yin Yang. Rainbow signature: One of the three nist post-quantum signature finalists.* https://www.pqcrainbow.org/, *2021.*

[Sen00] *Nicolas Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm.* IEEE Trans. Inf. Theory, *46(4):1193–1203, 2000. doi:10.1109/18.850662.*

[Sho97] *Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer.* SIAM J. Comput., *26(5):1484–1509, 1997. doi:10.1137/S0097539795293172.*

[Sto12] *Anton Stolbunov.* Cryptographic schemes based on isogenies. *PhD thesis, Norwegian University of Science and Technology, 2012.*

[TDJ+22] *Gang Tang, Dung Hoang Duong, Antoine Joux, Thomas Plantard, Youming Qiao, and Willy Susilo. Practical post-quantum signature schemes from isomorphism problems of trilinear forms. In Orr Dunkelman and Stefan Dziembowski, editors,* Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III, *volume 13277 of* Lecture Notes in Computer Science, *pages 582–612. Springer, 2022. doi:10.1007/978-3-031-07082-2_21.*

[TW05] *Patrick P. Tsang and Victor K. Wei. Short linkable ring signatures for e-voting, e-cash and attestation. In Robert H. Deng, Feng Bao, HweeHwa Pang, and Jianying Zhou, editors,* Information Security Practice and Experience, First International Conference, ISPEC 2005, Singapore, April 11-14, 2005, Proceedings, *volume 3439 of* Lecture Notes in Computer Science, *pages 48–60. Springer, 2005. doi:10.1007/978-3-540-31979-5_5.*

[Unr16] *Dominique Unruh. Computationally binding quantum commitments. In* Advances in Cryptology – Eurocrypt 2016, *pages 497–527. Springer, 2016.*

[Unr17] *Dominique Unruh. Post-quantum security of fiat-shamir. In* International Conference on the Theory and Application of Cryptology and Information Security, *pages 65–95. Springer, 2017.*

[VBC+19] *Javier A. Verbel, John Baena, Daniel Cabarcas, Ray A. Perlner, and Daniel Smith-Tone. On the complexity of "superdetermined" minrank instances.*

In Jintai Ding and Rainer Steinwandt, editors, Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers, *volume 11505 of* Lecture Notes in Computer Science, *pages 167–186. Springer, 2019. doi:10.1007/978-3-030-25510-7_10.*

[Wey97] *H. Weyl.* The classical groups: their invariants and representations, *volume 1. Princeton University Press, 1997.*

[Wil09a] *James B. Wilson. Decomposing p-groups via Jordan algebras.* Journal of Algebra, *322(8):2642–2679, 2009.*

[Wil09b] *R. Wilson.* The Finite Simple Groups, *volume 251 of* Graduate Texts in Mathematics. *Springer London, 2009.*

[YC04] *Bo-Yin Yang and Jiun-Ming Chen. All in the XL family: Theory and practice. In Choonsik Park and Seongtaek Chee, editors,* Information Security and Cryptology - ICISC 2004, 7th International Conference, Seoul, Korea, December 2-3, 2004, Revised Selected Papers, *volume 3506 of* Lecture Notes in Computer Science, *pages 67–86. Springer, 2004. doi:10.1007/11496618_7.*

[YEL+21] *Tsz Hon Yuen, Muhammed F. Esgin, Joseph K. Liu, Man Ho Au, and Zhimin Ding. Dualring: Generic construction of ring signatures with efficient instantiations. In Tal Malkin and Chris Peikert, editors,* Advances in Cryptology – CRYPTO 2021, *pages 251–281, Cham, 2021. Springer International Publishing.*

[YZ21] *Takashi Yamakawa and Mark Zhandry. Classical vs quantum random oracles. In* Annual International Conference on the Theory and Applications of Cryptographic Techniques, *pages 568–597. Springer, 2021.*

[Zha19] *Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors,* Advances in Cryptology – CRYPTO 2019*, pages 239–268, Cham, 2019. Springer International Publishing.*