# Solving the Tensor Isomorphism Problem for special orbits with low rank points: Cryptanalysis and repair of an Asiacrypt 2023 commitment scheme

Valerie Gilchrist[1], Laurane Marco[2], Christophe Petit[13], Gang Tang[43]

[1] Université Libre de Bruxelles, Brussels, Belgium
[2] EPFL, Lausanne, Switzerland
[3] University of Birmingham, Birmingham, United Kingdom
[4] University of Technology Sydney, NSW, Australia

**Abstract.** The Tensor Isomorphism Problem (TIP) has been shown equivalent to the matrix code equivalence problem, making it an interesting candidate on which to build post-quantum cryptographic primitives. These hard problems have already been used in protocol development. One of these, MEDS, is currently in Round 1 of NIST's call for additional post-quantum digital signatures.
In this work, we consider the TIP restricted to the orbits of a special class of tensors. The hardness of the decisional version of this problem is the foundation of a commitment scheme proposed by D'Alconzo, Flamini, and Gangemi (Asiacrypt 2023). We present polynomial-time algorithms for the decisional and computational versions of TIP for special orbits, which implies that the commitment scheme is not secure. The key observations of these algorithms are that these special tensors contain some low-rank points, and their stabilizer groups are not trivial.
With these new developments in the security of TIP in mind, we give a new commitment scheme based on the general TIP that is non-interactive, post-quantum, and statistically binding, making no new assumptions. Such a commitment scheme does not currently exist in the literature.

## 1 Introduction

Group actions have proven very useful in the transition to post-quantum cryptography. They provide a quantum-safe algebraic operation in certain instantiations, making it easy to use them as a replacement in previously classical schemes. Recently, isogenies have been the most well-studied example of a post-quantum group action [8, 13]. We can find, however, group actions from other hard problems, some of which have already been proposed for use in post-quantum cryptography. Such group actions include alternating trilinear forms [48], lattices [21], polynomials [41], linear codes [5, 16], and tensors [32].

* Authors listed in alphabetical order: see https://www.ams.org/profession/leaders/CultureStatement04.pdf.
Date of this document: 2024-06-14.

Grochow and Qiao [27] defined the tensor isomorphism class TI, that is, if a problem is polynomially equivalent to tensor isomorphism, it is TI-complete. In particular, the alternating trilinear form equivalence problem is TI-complete [28]. Based on this problem, a signature scheme ALTEQ [9] has recently been submitted to NIST as a round 1 candidate. The work of [27] shows that three problems, the matrix code equivalence problem, the trilinear form equivalence problem, and the tensor isomorphism problem (TIP) are all equivalent. Particularly given a 3-tensor with dimensions $l, m, n$, we can naturally represent it as a $[l \times m, n]$-matrix code. Thus, though our focus in this work will be on TIP, we will be inherently studying the security of all three of these problems. Two recent independent works [16,32] proposed using tensor isomorphism (or matrix code equivalence) for cryptographic purposes from the different algebraic structures of tensors and codes. Very recently, a digital signature scheme, MEDS [15], based on matrix code equivalence was submitted to the NIST standardization competition for post-quantum signatures as a round 1 candidate. The security of MEDS, that is, the algorithm of matrix code equivalence, has been studied in several works [15,16,18,39,44]. We proceed to briefly review the state-of-the-art algorithms for matrix code equivalence, and hence TIP.

*Algorithms solving matrix code equivalence.* There are three types of algorithms for matrix code equivalence. The first ones use algebraic methods, whereby matrix code equivalence is translated into solving systems of polynomial equations. Gröbner basis techniques are often employed for this approach, albeit with efficiency contingent upon the values of $l, m$ and $n$. Notably, efficiency diminishes significantly when these dimensions become slightly large. The second is the graph-theoretic algorithm [44], which involves transforming matrix codes into quadratic mappings, thereby facilitating the adaptation of algorithms designed for Quadratic Maps Linear Equivalence (QMLE) [10]. Nevertheless, this approach does not perform well when $l = m = n$. The last one is the Leon-like algorithm, which is an adaptation of the code equivalence problem algorithm on the Hamming metric. The basic idea comes from the observation that equivalence preserves the Hamming weight as well as the weight distribution of the codewords. Leon's [35] algorithm entails finding the set of codewords with minimal Hamming weight in two given codes, which can reveal enough information to recover the equivalence. Beullens [7] recently improved this algorithm by building two lists of codewords with a particular weight and then searching for collisions between them to recover the equivalence. This approach naturally translated to matrix code [16]: by creating two lists of matrices with low rank in the two given matrix codes one can then find the collision to recover the equivalence. Very recently, Narayanan, Qiao and Tang [39] further improved Beullens' algorithm and introduced a new invariant called "co-rank1 associated invariant" that avoids finding collisions by Gröbner basis.

In this work, we will consider TIP, but on special orbits. This problem was first introduced at ASIACRYPT 2023 by D'Alconzo, Flamini and Gangemi [20] in the context of a commitment scheme construction. While the general TIP considers tensors that are generated randomly, which we call *random tensors*,

the work of [20] restricts to the use of very particular tensors with a lot of structure, which we will call *unit tensors*. Their commitment scheme uses the orbits of two of these unit tensors, which they conjecture is still secure.

### 1.1 Our contributions

In this paper, we propose polynomial-time algorithms for the decisional and computational versions of TIP on special orbits. This implies that the commitment scheme proposed by D'Alconzo, Flamini and Gangemi in Asiacrypt 2023 is broken. We also propose a countermeasure to fix the commitment scheme. We summarize our contributions in what follows.

*Algorithms for tensor isomorphism on special orbits.* We present a polynomial-time algorithm for the TIP on special orbits. One observation is that some low-rank points exist for the tensors on these special orbits. This allows us to distinguish tensors on two different orbits, which implies that the decisional TIP is broken. Another observation is that the automorphism groups of these unit tensors are not trivial, that is, they have stabilizers. Therefore, we can avoid finding collisions and quickly recover an equivalent isomorphism which solves the computational TIP.

Regarding the commitment scheme proposed by D'Alconzo, Flamini and Gangemi, its security is based on the hardness of decisional TIP on these unit tensors. We implement our algorithms and use them to break the hiding property of the scheme, as well as recover the random values used when creating the commitment. These algorithms render the scheme completely insecure; see more details in Section 3.

*Repairing Asiacrypt 2023 scheme.* We provide a countermeasure to repair the commitment scheme from [20] using *random* (instead of unit) tensors. Our new scheme fills the gap of a post-quantum, non-interactive commitment scheme from non-transitive group actions, which was the aim of the Asiacrypt 2023 commitment scheme [20], and which is still missing in the literature. We stress that our new scheme makes no new assumptions, but only depends on already existing hard problems from random tensors. The scheme we propose is statistically binding and computationally hiding. We conclude by proposing a zero-knowledge proof of opening for our new commitment scheme.

*Organization.* In Section 2 we summarize the relevant concepts pertaining to group actions, tensors, commitment schemes, and the protocol we will be attacking from [20]. In Sections 3.1 and 3.2 we discuss low rank points and our attack on the hiding property of [20], and in Sections 3.3, 3.4, and 3.5, we compute stabilizers and give an attack that recovers the full secret. Finally, in Section 4, we give a repair on their commitment scheme that preserves the non-interactive structure and still has statistical binding, with a discussion on relevant zero-knowledge proofs in Section 4.2.

*Acknowledgements.* We thank the anonymous reviewers for their careful reading and several suggestions. We thank Youming Qiao for the helpful discussions and for referring us to [12]. Valerie Gilchrist is supported by a FRIA grant by the National Fund for Scientific Research (F.N.R.S.) of Belgium; Christophe Petit is partly supported by EPSRC through grant number EP/V011324/1; Gang Tang is partly supported by the Australian Research Council Linkage Projects LP220100332, Sydney Quantum Academy and EPSRC through grant number EP/V011324/1.

## 2 Preliminaries

In the rest of this work, we denote by $\mathbf{1}_A(x)$ the function taking value 1 if $x \in A$ and 0 otherwise. We write PPT to stand for Probabilistic Polynomial Time. We call a function *negligible*, written $\mathrm{negl}(\lambda)$, if its absolute value is asymptotically dominated by $O(x^{-n})$ for all $n > 0$.

### 2.1 Cryptographic group actions

Group actions have been of interest in cryptography because they sometimes provide an easy way to transition classical cryptosystems to post-quantum ones. The first well-studied instance of a group action used in cryptography was in works from Couveignes [17], and Rostovstev and Stolbunov [45], where they developed a Diffie-Hellman-like key exchange scheme from the isogeny group action.

The seminal work from Alamati, De Feo, Montgomery, and Patranabis in [1] generalizes the hard problems taken from the isogeny group action for use with any group action. In doing so, they establish standard definitions and notation that have since been upheld in the field, providing a valuable common ground for proceeding works. To begin, we recall some of these definitions that will be relevant to our work.

**Definition 1 (Group action).** *We say that a group $G$ acts on a set $X$ if there exists a map $\star : G \times X \to X$ such that :*

- *Identity: If $e$ is the identity element of $G$, then for any $x \in X$, $e \star x = x$;*
- *Compatibility: For any $g, h \in G$, and any $x \in X$, $(gh) \star x = g \star (h \star x)$.*

*Such a group action can be denoted $(G, X, \star)$.*

We call a group action *transitive* if for any two elements $x, y \in X$, there exists a group element $g \in G$ mapping $x$ to $y$, i.e. $y = g \star x$. We say a group action is *free* if for each $g \in G$, $g$ is the identity element if and only if there exists an element $x \in X$ such that $x = g \star x$. A group action is called *regular* if it is both transitive and free.

**Definition 2 (Orbit).** *The* orbit*, $O_x$, of an element $x \in X$, is all the $y \in X$ for which there exists an element mapping $x$ to $y$ i.e $O_x = \{y \in X | \exists g \in G, y = g \star x\}$.*

We can reformulate transitivity in terms of orbits : an action is transitive if for all $x, y \in X$, we get that $O_x = O_y$.

Since we are considering group actions in the context of cryptography, we will only be concerned with *effective group actions*.

**Definition 3 (Effective group action (EGA)).** *We call a group action* $\star : G \times X \to X$ effective *if the following properties hold :*

- *$G$ is finite and there exists a* PPT *algorithm for the following operations : group operation, computation of inverses, membership testing, equality testing and sampling.*
- *$X$ is finite and there exists a* PPT *algorithm for membership testing and for computing a unique representation of elements in $X$.*
- *There exists a distinguished element $x_0 \in X$ such that its bit-string representation is known. We call this point the* origin.
- *$\star$ can be computed efficiently for any $g \in G, x \in X$.*

When creating cryptography from group actions, we will want to be able to actually evaluate the group action, hence we will limit ourselves to EGAs. In some instances where only a portion of the group or set can be evaluated efficiently, we can restrict to such a subset. This is called a *Restricted Effective Group Action (REGA)*.

We can impose some additional properties to be fulfilled by the group action in order to build specific cryprographic constructions. We recall a definition from [20] that will be useful later on in this work.

**Definition 4 (Decisional Group Action Inverse Problem ).** *Let $(G, X, \star)$ be a group action, and $t_0, t_1 \in X$ lie in distinct orbits. The decisional Group Action Inverse Problem (dGA-IP) game for $(G, X, \star)$ is described in Figure 1. For an adversary $\mathcal{A}$ against the* dGA-IP *game, we define the advantage* Adv *as follows :*

$$\mathrm{Adv}(\mathcal{A}) = \Pr[1 \leftarrow \mathsf{dGA\text{-}IP}(\mathcal{A})] - \frac{1}{2}.$$

### 2.2 Tensors over finite fields

We recall some results on tensors, following the notations introduced in [20].

Fix a finite field $\mathbb{F}_q$ for a prime $q$, and $\ell, m, n$ positive integers. Then given bases $\{e_i\}_{i=1}^{l}$, $\{f_i\}_{i=1}^{m}$, $\{g_i\}_{i=1}^{n}$ of $\mathbb{F}_q^l$,$\mathbb{F}_q^m$,$\mathbb{F}_q^n$, respectively, a 3-tensor $v \in \mathbb{F}_q^l \otimes \mathbb{F}_q^m \otimes \mathbb{F}_q^n$ is defined as follows

$$v = \sum_{i=1}^{\ell} \sum_{j=1}^{m} \sum_{k=1}^{n} v(i,j,k) e_i \otimes f_j \otimes g_k.$$

Alternatively a *3-tensor*, $v$, can be represented as a 3-way array of field elements

$$v = [[v(i,j,k)]]_{i,j,k=1}^{l,m,n} \in \mathbb{F}_q^{l \times m \times n}.$$

$\underline{\mathsf{dGA\text{-}IP}(\mathcal{A})}$

$c, b \xleftarrow{\$} \{0,1\}$
$g, g' \xleftarrow{\$} G$
$s \leftarrow g \star t_c$
**if** $b = 1$ **then**
    $t \leftarrow g' \star s$
**end if**
**if** $b = 0$ **then**
    $t \leftarrow g' \star t_{1-c}$
**end if**
$b' \leftarrow \mathcal{A}(s, t)$
**return** $b' = b$

**Fig. 1.** The dGA-IP game

In the same way that matrices encode bilinear forms, 3-tensors encode trilinear forms. More precisely, given a 3-tensor $v$, one can associate the 3-linear form $L_v : \mathbb{F}_q^l \times \mathbb{F}_q^m \times \mathbb{F}_q^n \to \mathbb{F}_q$ defined as

$$L_v(x^1, x^2, x^3) = \sum_{i_1=1}^{l} \sum_{i_2=1}^{m} \sum_{i_3=1}^{n} v(i_1, i_2, i_3) x_{i_1}^1 x_{i_2}^2 x_{i_3}^3.$$

For the rest of this work, we restrict to the case where $l = m = n$. We will also only be concerned with 3-tensors, even if not explicitly stated, since the work from [27] gives an equivalence between isomorphism problems on 3-tensors and higher dimensional $d$-tensors. This makes our study of 3-tensors an optimal instance of these problems. Let $\mathbf{V}$ be the tensor space $\mathbf{V} = \mathbb{F}_q^n \otimes \mathbb{F}_q^n \otimes \mathbb{F}_q^n$, and we will use the canonical basis $\{e_i\}_{i=1}^n$ on $\mathbb{F}_q^n$ to construct tensors.

*Tensor rank.* An essential invariant of tensors that we will be considering is their *rank*.

**Definition 5 (Rank 1 tensor).** *A tensor, $v \in \mathbf{V}$, is said to have* rank 1 *if it can be written as $v = a \otimes b \otimes c$, for $a, b, c \in \mathbb{F}_q^n$.*

**Definition 6 (Rank of a tensor).** *The* rank *of a tensor, $v \in \mathbf{V}$, is the minimal $r$ such that we can write $v$ in the form $v = \sum_{i=1}^{r} w_i$, where $\{w_i\}_{i=1}^r \subset \mathbf{V}$ is a set of rank 1 tensors.*

*Problem 1 (Computational Tensor Rank Problem).* Given a tensor $v \in \mathbf{V}$, compute $\mathrm{rank}(v)$.

It has been shown that Problem 1 is NP-hard [29, 30, 46].

It is shown in [30] that other tensor related problems are NP-hard. These include the decisional and computational versions of problems investigating eigenvalues, singular values, and spectral norms of tensors. These problems, however, are beyond the scope of this paper.

*Group action on tensors.* We can obtain further hard problems from tensors by defining a group action on them.
More concretely, the tensors are being acted on by the group $G = \mathrm{GL}(n,q) \times \mathrm{GL}(n,q) \times \mathrm{GL}(n,q)$, where $\mathrm{GL}(n,q)$ denotes the general linear group of degree $n$ over $\mathbb{F}_q$, in the following way:

$$\star : G \times \mathbf{V} \to \mathbf{V},$$

$$\Big((A,B,C), \sum_{i,j,k} v(i,j,k) e_i \otimes e_j \otimes e_k\Big) \mapsto \sum_{i,j,k} v(i,j,k) Ae_i \otimes Be_j \otimes Ce_k, \quad (1)$$

for $v(i,j,k) \in \mathbb{F}_q$.

**Lemma 1.** *Let $(A,B,C) \in G$, and $v \in \mathbf{V}$. Then $rank((A,B,C)\star v) = rank(v)$. In other words, $(\star)$ preserves rank.*

*Proof.* We can decompose $w := (A,B,C)\star v$ into a sum as shown in Equation 1, giving us an upper bound on its rank. Thus $rank(w) \leq rank(v)$.

Now consider the tensor $(A^{-1}, B^{-1}, C^{-1}) \star w$, which again can be written as a sum whose index is bounded above by $rank(w)$. This gives us that $rank((A^{-1}, B^{-1}, C^{-1})\star w) \leq rank(w)$. We notice, however, that $(A^{-1}, B^{-1}, C^{-1})\star w = v$, thus we get that $rank(v) \leq rank(w)$ and from before, $rank(w) \leq rank(v)$. Hence $rank(v) = rank(w)$.

□

Given this group action on tensors, we can consider two additional potentially hard problems.

*Problem 2 (Decisional Tensor Isomorphism Problem).* Given two tensors $v_0, v_1 \in \mathbf{V}$, decide whether there exists $(A,B,C) \in G$ such that $(A,B,C) \star v_0 = v_1$.

*Problem 3 (Computational Tensor Isomorphism Problem).* Given two tensors $v_0, v_1 \in \mathbf{V}$, such that $(A,B,C) \star v_0 = v_1$ for some $(A,B,C) \in G$, compute $(A,B,C)$.

We can rephrase these problems in terms of *orbits*, from Definition 2 : Problem 2 asks to determine whether two tensors belong to the same orbit; Problem 3 asks to determine the isomorphism mapping two elements from the same orbit.

### 2.3 Commitment schemes

Commitment schemes have many real-world use cases, finding applications in multiparty computation, zero-knowledge proofs, and coin tossing [24, 33, 36, 40, 43].

A commitment scheme is used when a sender wants to commit to some value, $m$, without initially revealing it to the receiver. Instead, the sender will commit to $m$ by computing a commitment, $c$, that depends on $m$, and send it to the receiver. At a later time, the sender will reveal $m$, and the receiver should be

able to verify that $c$ was computed using $m$. The key security properties here are *hiding*, which means $c$ should reveal nothing about $m$; and *binding*, which means no other value $m' \neq m$ should be able to open $c$. We proceed to give formal definitions for a commitment scheme and its security properties.

**Definition 7 (Commitment scheme).** *A commitment scheme is defined by a message space, $\mathcal{M}$, randomness space, $\mathcal{R}$, a commitment space, $\mathcal{C}$, and a tuple of algorithms,* $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify})$*, defined in the following way :*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$*: This* PPT *algorithm takes as input a security parameter $\lambda$ in unitary form and returns some public parameters* pp *which will be implicitly given as inputs to the proceeding two algorithms.*
- $\mathsf{Commit}(m, r) \to c$ *: This* PPT *algorithm takes as input a message $m \in \mathcal{M}$ and randomness $r \in \mathcal{R}$ and outputs a commitment value $c \in \mathcal{C}$.*
- $\mathsf{Verify}(c, (m, r)) \to 0/1$*: This is a deterministic algorithm which, given as input $c \in \mathcal{C}$, $(m, r) \in \mathcal{M} \times \mathcal{R}$ returns 1 if $\mathsf{Commit}(m, r) = c$ and 0 otherwise.*

In the case of a *bit commitment scheme*, the committed messages are restricted to two possible values, 0 or 1.

**Definition 8 (Hiding Security).** *A commitment scheme* $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify})$ *is* hiding *if for any adversary, $\mathcal{A}$, the advantage,* Adv*, is negligible, where* Adv *is defined as*

$$\mathrm{Adv} = |\Pr[1 \leftarrow \mathsf{Hiding}(\mathcal{A})] - 1/2|$$

*and the* Hiding *game is defined in Figure 2. In this figure, we take $m_0, m_1, st$ to be two messages and a state chosen by the adversary $\mathcal{A}$ for use in the game.*

$\underline{\mathsf{Hiding}(\mathcal{A})}$
1: $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
2: $m_0, m_1, st \leftarrow \mathcal{A}(\mathsf{pp})$
3: $b \xleftarrow{\$} \{0,1\}, r \xleftarrow{\$} \mathcal{R}$
4: $c \leftarrow \mathsf{Commit}(m_b, r)$
5: $b' \leftarrow \mathcal{A}(c, st)$
6: **return** $b' = b$

$\underline{\mathsf{Hiding}_{\mathsf{Bit}}(\mathcal{A})}$
1: $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
2: $b \xleftarrow{\$} \{0,1\}, r \xleftarrow{\$} \mathcal{R}$
3: $c \leftarrow \mathsf{Commit}(b, r)$
4: $b' \leftarrow \mathcal{A}(c)$
5: **return** $b' = b$

**Fig. 2.** The Hiding and $\mathsf{Hiding}_{\mathsf{Bit}}$ game

*Remark 1.* Note that in the case of a bit commitment scheme, we have $\mathcal{M} = \{0,1\}$, and in the Hiding game the adversary does not choose $m_0, m_1$ and does not need to provide a state $st$, rather the bit committed is the one sampled by the game, i.e. one can remove Line 2 in Figure 2 and set $m_b = b$, as depicted on the right-hand side of Figure 2 by the $\mathsf{Hiding}_{\mathsf{Bit}}$ game.

**Definition 9 (Binding Security).** *A commitment scheme* (Setup, Commit, Verify) *is* binding *if for any adversary* $\mathcal{A}$ *the advantage* Adv *is negligible, where* Adv *is defined as*

$$\text{Adv} = \Pr[1 \leftarrow \mathsf{Binding}(\mathcal{A})]$$

*and the* Binding *game is as defined in Figure 3.*

$\underline{\mathsf{Binding}(\mathcal{A})}$
1: $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
2: $m_0, m_1, r_0, r_1 \leftarrow \mathcal{A}(\mathsf{pp})$
3: **return** $(m_0 \neq m_1) \wedge \mathsf{Commit}(m_0; r_0) = \mathsf{Commit}(m_1; r_1)$

**Fig. 3.** The Binding game

*Remark 2.* We can make the hiding and binding definition more precise by quantifying the complexity of the adversary as well as its advantage. When $\mathcal{A}$ has unbounded complexity and $\text{Adv} = 0$, the commitment scheme is said to be *perfectly* binding/hiding. When $\mathcal{A}$ has unbounded complexity and $\text{Adv} = \text{negl}(\lambda)$, it is called *statistically* binding/hiding. When $\mathcal{A}$ is PPT , and $\text{Adv} = \text{negl}(\lambda)$ it is called *computationally* binding/hiding.

### 2.4 Asiacrypt 2023's commitment scheme from tensors

While there is already literature about commitment schemes [38,42], developing post-quantum equivalents is necessary in order to curb the risk of future quantum attacks. Many post-quantum commitments exist from lattice-based assumptions with recent constructions such as [2,6,22], code-based assumptions [31,37], isogeny-based assumptions [47] and *transitive* group actions [11, 14, 32]. *Non-transitive* group actions, however, are less restrictive and arise naturally, making them interesting for use in commitment schemes. In [20], at Asiacrypt 2023, D'Alconzo, Flamini, and Gangemi describe a general bit commitment framework from post-quantum non-transitive group actions that they call GACE (Group Action with Canonical Element), and then give an instance using tensors in Section 6.2 of their work.

In particular, they take inspiration from the fact that it is widely believed to be hard to compute the rank of a random tensor (see Section 2.2), and proven NP-hard [29,30,46]. In the paper they provide a method of constructing a distinguished element for any rank, such that given the rank and some additional data it is easy to verify the correctness of the original claim. Let $b \in \{0, \dots n-1\}$, then their distinguished element is

$$t_b := \sum_{i=1}^{n-b} e_i \otimes e_i \otimes e_i. \tag{2}$$

We will refer to the $t_b$ as *unit tensors*, following the terminology from [12].

In their commitment scheme they use $t_b$ to create either an $n$-rank tensor or an $(n-1)$-rank tensor, thereby encoding a bit $b \in \{0,1\}$. They then sample an element $(A, B, C) \in G = \mathrm{GL}(n,q) \times \mathrm{GL}(n,q) \times \mathrm{GL}(n,q)$, and compute the commitment $c = (A, B, C) \star t_b$ via the group action described in Equation 1. The sender sends the commitment $c$ to the receiver. When the sender is ready, they will reveal $(A, B, C), b$. The receiver can then easily check that $c = (A, B, C) \star t_b$, and so has rank $n - b$.

*Parameters.* There are no parameters given for the commitment scheme from [20], and it does not rely on a pure form of the tensor isomorphism problem, so parameter choice for this scheme is not obvious. Though our attacks run in polynomial time complexity, we would like a concrete example instance of the commitment scheme on which to test the algorithms outlined in Section 3. Thus, we will consider the parameters outlined in MEDS [15], a post-quantum signature from matrix code equivalences, submitted to the NIST competition for post-quantum signatures. Their proposed choices of vector dimension and field size, $(n, q)$, are $(14, 4093), (22, 4093), (30, 2039)$.

The matrix code equivalence problem is polynomially equivalent to the general trilinear form equivalence problem and the tensor isomorphism problem as noted in [27], meaning parameter sizes should be the same across all of these hard problems.

### 2.5 The MinRank problem

In Section 3 we will be exploring how low rank points can affect the security of some tensor-based hard problems. For this, we will need a way of solving a MinRank instance. While some more straightforward approaches to MinRank, such as [34], would suffice for computing low rank points, we choose to use a more general algorithm to eliminate the chance of using different rank tensors as a countermeasure to our attacks later on. Here we summarize the relevant results from [4], the current state-of-the-art for this problem. The authors of this work revise the algebraic approach to MinRank, and thus completely avoid the use of a Gröbner basis computation for some parameters, keeping all the equations linear. When applying this approach to three NIST candidate key exchange schemes, they were able to improve upon or match the state-of-the-art attacks.

The MinRank problem is as follows:

*Problem 4 (MinRank problem).* Given an integer $r \in \mathbb{N}$, and $k$ matrices $M_1, \dots M_k \in \mathbb{F}_q^{m \times n}$, determine field elements $x_1, \dots x_k$ not all zero, such that

$$\mathrm{rank}\Big(\sum_{i=1}^{k} x_i M_i\Big) \leq r.$$

In previous works [3], the MinRank problem was solved by creating equations using *coefficient* variables and *support* variables depending on maximal minor equations. This system of equations would then be solved using a Gröbner basis algorithm. In [4], they instead define even more bilinear equations using the coefficient and support variables. So many, in fact, that they are sufficient for solving the entire problem, thereby avoiding the Gröbner basis computation in many cases.

*The algorithm.* We are given $n$ matrices, $M_1, \dots M_n$, of size $n \times n$, and a target rank $r$. We want to find $\{x_i\}_{i=1}^n$ such that $\mathbf{M} := \sum x_i M_i$ has rank $r$. So the entries of $\mathbf{M}$ are linear expressions in the variables $x_i$. When $\mathbf{M}$ has rank $r$, we can write it as $\mathbf{M} = SR$, where columns of the $n \times r$ matrix $S$ form a basis for the column-space of $\mathbf{M}$, and the $r \times n$ matrix $R$ contains a basis for the row-space of $\mathbf{M}$ . The entries in $S$ and $R$ are referred to as the *support* and *coefficient* variables, respectively. The matrix $R$ will necessarily have full rank $r$. Hence, if we add a row from $\mathbf{M}$ to it, all of its maximal minors will be vanishing. These vanishing maximal minors give us algebraic equations depending on the $x_i$ and the entries of $R$. We can do this for each row of $\mathbf{M}$, obtaining enough equations to linearize and solve for the $x_i$.

### 2.6 Sigma protocols

We give below an informal definition of a sigma protocol that will be of use in Section 4.2.

**Definition 10 (Sigma protocol).** *A sigma protocol $\Sigma$ for a relation $\mathcal{R}$ is a protocol between a prover $P$ and a verifier $V$. It has input $(x, w) \in \mathcal{R}$ where $x$, the statement, is a common input and $w$, the witness, is private to $P$, and it must satisfy the following :*

- *$\Sigma$ is a three-move protocol, where $P$ sends the first message $a$, $V$ replies with some string $e$, $P$ send a final message $z$ upon which $V$ wither accepts or rejects.*
- *It is complete i.e. if $P, V$ follow the protocol on input $(x, w)$ with $(x, w) \in \mathcal{R}$ then $V$ always accepts.*
- *Special soundness: there exists a polynomially bounded algorithm $\mathcal{E}$ called* extractor *such that for any $x$, if $(x, a, e, z)$ and $(x, a, e', z)$ are two accepting views for $\mathcal{V}$ such that $e \neq e'$ then $\mathcal{E}(x, a, e, z, e', z')$ yields $w$ such that $R(x, w)$.*
- *Special honest-verifier zero-knowledge : there exists a polynomially bounded algorithm $\mathsf{Sim}$ called* simulator *such that for any $x \in L$ and $e$, the transcript $(a, e, z)$ of the interaction $\mathcal{P} \overset{x}{\leftrightarrow} \mathcal{V}$ conditioned to $e$ has the same distribution as $\mathsf{Sim}(x, e)$.*

A sigma-protocol can be turned into a non-interactive zero-knowledge proof using the Fiat-Shamir transform [23].

## 3 Solving Tensor Isomorphism Problems in $t_b$ orbits

In this section we focus on attacking the commitment scheme from [20], which uses special cases of tensor hard problems. We begin in Section 3.1 by establishing some background on the rank of points in tensors. This will help us in Section 3.2, where we focus on attacking a variant of the Decisional Tensor Isomorphism Problem (DTI, see Problem 2) present in their commitment scheme, and provide a polynomial time attack on it. We then describe some elements of the stabilizer group of the family of tensors $\{t_b : b \in \mathbb{Z}_n\}$ in Section 3.3. Lastly, in Sections 3.4 and 3.5, we use these findings to attack their variant of the Computational Tensor Isomorphism Problem (CTI, see Problem 3).

All the algorithms and experiments described in the proceeding sections were coded in Magma, and can be found at

https://github.com/vgilchri/tensor-group-action .

### 3.1 Computing the rank of points

In the case of tensors, we can first choose to view a 3-tensor, $g \in \mathbf{V}$, as a list of $n$ matrices $G_1, \dots G_n \in \mathbb{F}_q^{n \times n}$. Given $u, v, w \in \mathbb{F}_q^n$, we represent $u \otimes v \otimes w$ as $[G_1, ..., G_n]$ where $G_i = u_i \cdot (v \cdot w^T)$. Note, there are two additional ways to do this, which essentially changes along which axis you are "slicing". While the choice of this axis is not important, choosing one and staying consistent is. We highlight our chosen representation in the following example.

*Example 1.* Suppose we are working over $\mathbb{F}_5^3$. We would like to compute the tensor $u \otimes v \otimes w$, where $u = [4, 1, 2], v = [3, 0, 1], w = [2, 4, 0]$.

We proceed by first expanding the matrix $v \cdot w^T$ :

$$v \cdot w^T = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 0 & 0 \\ 2 & 4 & 0 \end{bmatrix}.$$

Now we multiply this matrix by each entry of $u$, storing them in a list as we go and we obtain the following :

$$\begin{bmatrix} 4 & 3 & 0 \\ 0 & 0 & 0 \\ 3 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 0 \\ 0 & 0 & 0 \\ 2 & 4 & 0 \end{bmatrix}, \begin{bmatrix} 2 & 4 & 0 \\ 0 & 0 & 0 \\ 4 & 3 & 0 \end{bmatrix}.$$

Using this representation of tensors, we can define the rank of a *point* in a tensor.

**Definition 11 (Rank of a point).** *Let $g \in \mathbf{V}$ be a tensor, written in the form $g = [G_1, \dots G_n]$. Then the rank of a point $u := [u_1, \dots u_n] \in \mathbb{F}_q^n$ in $g$ is exactly the matrix rank of $u_1 G_1 + \dots u_n G_n$.*

**Lemma 2.** *The set of rank 0 points of a tensor, $v \in \mathbf{V}$, is a vector space.*

*Proof.* Associativity, commutativity, and distributivity are all inherited from $\mathbb{F}_q^n$. The zero-vector in $\mathbb{F}_q^n$ is trivially a rank 0 point in any tensor, and so serves as the additive identity element for the vector space. Furthermore, we have closure under additive inverses and scalar multiplication since multiplication by a non-zero scalar does not affect the rank of a matrix. Lastly, given two rank 0 points $u$ and $w$, we get that $u + w$ is a rank 0 point, since the sum of rank 0 matrices will have rank 0 as well. □

Computing the rank of points in a tensor will be useful to us later on. In the rest of this paper, when talking about the rank of points, we will consider points up to scalar multiplication, unless stated otherwise.

**Lemma 3.** *Let $v \in \mathbf{V}$ be a tensor. Denote $L_k(v) := \{u \in \mathbb{F}_q^n : rank_v(u) = k\}$ to be the set of rank $k$ points in $v$. Now let $(A, B, C) \in G$ be an isomorphism, and $w = (A, B, C) \star v$. Then $|L_k(v)| = |L_k(w)|$.*

*In particular, we get a bijection that sends $u \in L_k(v)$ to $uA^{-1} \in L_k(w)$.*

*Proof.* We observe that, given a tensor $v$, the action of $B$ and $C$ does not affect the rank of a point, meaning that a point $u$ of rank $r$ in $v$ will also be of rank $r$ for $(I, B, I) \star v$, and $(I, I, C) \star v$. This follows straightforwardly from the fact that given $v = [T_1, ..., T_n]$, we can write the action of $(I, B, I)$ as $(I, B, I) \star v = [BT_1, ..., BT_n]$ and that of $(I, C, I)$ as $(I, I, C) \star v = [T_1C^T, ..., T_nC^T]$ with $^T$ denoting the transpose.
We observe that only the action of $A$ has some influence on the rank of points, and $A$ sends a rank $r$ point $u$ in $v$ to the rank $r$ point $u' = uA^{-1}$ in $(A, I, I) \star v$, where $\star$ is as defined in Equation 3.3.
Combining these results, we obtain a one-to-one correspondence between the rank $r$ points in $v$, and the rank $r$ points in $w = (A, B, C) \star v$, with an explicit mapping, which proves the statement. □

When considering tensors, Lemma 3 tells us that the number of points of fixed rank in a tensor is preserved via the group action ($\star$) from Equation 3.3. Computing a group action in this case amounts to transforming the bases of $\mathbb{F}_q^n$ being used to compute the tensor, to new bases, via linear transformations (recall we require three bases of $\mathbb{F}_q^n$). Hence, the overall structure or rank of the tensor does not change.

Rank 0 points in tensors, when considering a tensor as a list of matrices, means that there exists some linear dependency between these matrices. Transforming the bases of $\mathbb{F}_q^n$ preserves this dependency, regardless of the exact changes of bases being used. Solving for these points can be done with Gaussian elimination, as we will see in Section 3.2. Solving for higher rank points will amount to a MinRank problem, described in Section 2.5.

### 3.2 Hiding attack

We now describe an attack that breaks the hiding property of the commitment scheme described in [20]. That is, given a committed value $c = \mathsf{Commit}(b, r)$ for

a given bit $b$, we recover $b$ by solving a system of linear equations, therefore disproving the security claims made in [20].

Let us recall the problem we are attacking:

*Problem 5.* Given $c = (A, B, C) \star t_b$ with $t_b = \sum_{i=1}^{n-b} e_i \otimes e_i \otimes e_i$, $b \in \{0, \dots n-1\}$, $(A, B, C) \in G$ sampled randomly, and $\star$ is as defined in Equation 3.3, recover $b$.

Note, this problem is slightly more general than the commitment scheme from [20], since we are considering any $b \in \{0, 1, \dots n-1\}$. From Problem 5, we see that $c$ is precisely the tensor $t_b$ with an isomorphism acting on it, as described in Lemma 3.

**Lemma 4.** *Let $t_b = \sum_{i=1}^{n-b} e_i \otimes e_i \otimes e_i$, then the vector space of rank 0 points has dimension $b$.*

*Proof.* Lemma 2 immediately tells us that the set of rank 0 points is a vector space. It remains to prove the dimension.

Let us write $t_b$ as an $n$ list of $n \times n$ matrices, which we denote $T_b^1, \dots T_b^n$. These matrices can be described in the following way :

$$(T_b^k)_{ij} = \begin{cases} 1 \text{ if } i = j = k \text{ and } k \leq n - b \\ 0 \text{ otherwise} \end{cases}$$

We know that the rank of a point $u \in \mathbb{F}_q^n$ of $t_b$ is the rank of the matrix $U_b = u_1 T_b^1 + .. + u_n T_b^n$.
Using the above definition of the $T_b^i$, $i = 1, .., n$ we observe that $U_b$ will be a diagonal matrix with entries

$$(U_b)_{ii} = \begin{cases} u_i, \text{ for } i = 1, ..., n - b \\ 0 \text{ otherwise} \end{cases} .$$

Suppose $u \neq 0 \in \mathbb{F}_q^n$, then $U_b$ will consist of a diagonal matrix of dimension $(n-b) \times (n-b)$ with the remaining $b$ rows/columns being completely zeros. This upper bounds the rank to be at most $n - b$. For example, $U_0$ will have rank at least 1 when $u$ is non-zero, and therefore no non-trivial rank 0 points and thus the subspace has dimension 0. Similarly $U_1$ will have one zero column so rank at most $n - 1$ and will always have a rank 0 point (up to scalar multiplication), in the form $u_1 = ... = u_{n-1} = 0$ and $u_n$ any non-zero element of $\mathbb{F}_q$.

More generally, for $t_b$ where $b > 0$, letting $e_i$ denote the canonical basis element of $\mathbb{F}_q^n$, we can define the basis $B$ generating the vector space of rank 0 points in $t_b$ as

$$B := \{e_n, \dots e_{n-b+1}\}.$$

□

We focus for a moment on the case $b \in \{0, 1\}$. Using Lemma 4, we have that $t_1$ has exactly one rank 0 point, and that $t_0$ has no rank 0 points. Combining

this with Lemma 3 shows that we can use the existence of a rank 0 point as a distinguisher on $c$ (as defined in Problem 5). We can find the rank 0 points of $c$ by solving the system of $n^2$ linear equations

$$\alpha_1 G_1 + ... + \alpha_n G_n = 0, \tag{3}$$

for $\alpha_i \in \mathbb{F}_q$, $i = 1, ..., n$. If such a solution exists, then $b = 1$, else $b = 0$. For the more general case of $b \in \{0, 1, \dots n-1\}$, the dimension of the solution space will be exactly $b$.

While more complex algorithms could be used to solve this system, Gaussian elimination is sufficient for our purposes. The challenge here is selecting a set of $n$ linearly independent equations from the $n^2$ possible choices. In the worst case this gives an upper bound on the number of operations of $O(n^4)$. This proves Theorem 1.

**Theorem 1.** *There exists an algorithm that solves Problem 5 using $O(n^4)$ operations.*

For all parameters highlighted in Section 2.4 the attack runs in under a second on a laptop. While running these experiments we found that sampling the first $n$ equations was sufficient for correctly solving the system. While we would not expect this to be true for a random set of $n^2$ equations, this could be due to the structure of the tensor which is not random. So heuristically, we expect a complexity of $O(n^3)$. Note that Theorem 1 always returns a correct answer to Problem 5.

To this end, we give a second probabilistic algorithm that also requires $O(n^3)$ operations but uses a slightly different approach to random sampling.

**Theorem 2.** *There exists a probabilistic algorithm that solves Problem 5 using $O(n^3)$ operations.*

*Proof.* Recall from Problem 5, given $c = (A, B, C) \star t_b$, we would like to recover $b$.

We begin by restricting to the case $b \in \{0, 1\}$. Consider a random point $u \in \mathbb{F}_q^n$. Then $u$ has rank $n$ in $t_0$ and rank $n-1$ in $t_1$ with very high probability. Namely $u$ will have rank $n$ in $t_0$ with probability $(q-1)^n/q^n$ since there can be no zeros in this point. Similarly, in $t_1$, one of the entries can be free but the rest must be non-zero, so we get probability $(q-1)^{n-1}/q^{n-1}$. By Lemma 3, these probabilities will translate to $c$. Thus, it suffices to compute the rank of $u$ in the commitment, $c$, to recover $b$ with overwhelming probability, for large $q$. This approach generalizes to $b \in \{0, 1, \dots n-1\}$, in the obvious way.

This algorithm consists of computing a random linear combination of $n$ square matrices, and computing the sum's rank over the finite field $\mathbb{F}_q$. Using Gaussian elimination, we can do this in $O(n^3)$ complexity. □

*Remark 3.* This attack method works because of how structured $t_b$ is. For a random tensor we do not expect to have any rank 0 point (this will be shown in Lemma 12). For this reason, our distinguishing attack is not expected to have

any impact on the general case of Problem 2 (the Decisional Tensor Isomorphism Problem), which asks an attacker to determine whether two given tensors are isomorphic. For the same reason, we also do not claim any improvement on Problem 3 (the Computational Tensor Isomorphism Problem). To date, both of these problems are believed to be cryptographically hard in the general setting, and we will use them in Section 4 to build a new commitment scheme from random tensors.

### 3.3 Stabilizer subgroup of $t_b$

We now investigate some elements belonging to the stabilizer group of tensors, and construct the entire stabilizer group of $t_0$. These findings will be of use to us in an attack in later sections.

*Stabilizers for all tensors.* Recall, we define $v \in \mathbf{V}$ as

$$v := \sum_{i,j,k=1}^{n} v(i,j,k) e_i \otimes e_j \otimes e_k,$$

and we are considering isomorphisms $(A, B, C) \in G$ that act on $v$ via the group action, $(\star)$, defined as

$$(A, B, C) \star \sum_{i,j,k} v(i,j,k) e_i \otimes e_j \otimes e_k = \sum_{i,j,k} v(i,j,k) Ae_i \otimes Be_j \otimes Ce_k.$$

**Lemma 5.** *Let $\lambda_a, \lambda_b, \lambda_c \in \mathbb{F}_q$ be such that $\lambda_a \lambda_b \lambda_c = 1$. Then for all $v \in \mathbf{V}$, we have that $(\lambda_a I_n, \lambda_b I_n, \lambda_c I_n) \star v = v$.*

The proof of this lemma can be done straightforwardly using the definition of the group action from Equation 3.3. We reserve the details for Appendix A.

*Stabilizer subgroup of $t_b$.* Now we focus our attention to $t_b$, which can be written as

$$t_b := \sum_{i=1}^{n-b} e_i \otimes e_i \otimes e_i.$$

In the following two lemmas we formalize how diagonal matrices and permutation matrices can be used to create stabilizing elements for $t_b$, and more specifically, for $t_0$.

**Lemma 6.** *Suppose $\Lambda_A, \Lambda_B, \Lambda_C$ are three diagonal $n \times n$ matrices over $\mathbb{F}_q$ such that $\Lambda_A \Lambda_B \Lambda_C = I_n$. Then $(\Lambda_A, \Lambda_B, \Lambda_C) \star t_b = t_b$, i.e. $(\Lambda_A, \Lambda_B, \Lambda_C)$ is in the stabilizer subgroup of $t_b$.*

*Proof.* We claim that $(\Lambda_A, \Lambda_B, \Lambda_C)$ is an element of the stabilizing group of $t_b$. We can see this explicitly as

$$\begin{aligned}(\Lambda_A, \Lambda_B, \Lambda_C) \star t_b &= \sum_{i,j,k=1}^{n-b} \sum_{s=1}^{n-b} \Lambda^A_{is}\Lambda^B_{js}\Lambda^C_{ks} e_i \otimes e_j \otimes e_k \text{ (the matrices are diagonal)} \\ &= \sum_{i=1}^{n-b} \Lambda^A_{ii}\Lambda^B_{ii}\Lambda^C_{ii} e_i \otimes e_i \otimes e_i \\ &= \sum_{i=1}^{n-b} e_i \otimes e_i \otimes e_i = t_b.\end{aligned}$$

□

*Additional stabilizer elements for $t_0$.* To explore how permutation matrices can be used to build stabilizing elements, we first focus on the case of $b = 0$. In what follows, we use $S_n$ to denote the symmetric group of integers $\{1, \dots n\}$.

**Lemma 7.** *Let $\sigma \in S_n$, and let $P_\sigma$ be the associated permutation matrix given by $P_\sigma = (p_{ij})_{i,j=1}^n$ and $p_{ij} = \mathbf{1}_{i=\sigma(j)}$. Then $(P_\sigma, P_\sigma, P_\sigma) \star t_0 = t_0$, i.e. $(P_\sigma, P_\sigma, P_\sigma)$ is in the stabilizer subgroup of $t_0$.*

*Proof.* Consider any permutation matrix $P_\sigma$. When applied consistently to the bases being used to construct $t_0 = \sum_{s=1}^n e_s \otimes e_s \otimes e_s$, it simply permutes the order in which the summation is performed, resulting in the same tensor $t_0$. We reserve the explicit calculations showing this equality for Appendix A. □

In [12, Proposition 4.1], they state that the entire stabilizer group of $t_0$, defined over the complex numbers, is exactly the semi-direct product of the diagonal matrices from Lemma 6 and the set of permutation matrices $P_\sigma$ for $\sigma \in S_n$. This is consistent with our findings over finite fields, and was further corroborated via our experiments.

*Additional stabilizer elements for $t_1$.* We now turn to the study of the stabilizer elements specific to $t_1$.

**Lemma 8.** *Let $\sigma \in S_{n-1}$, and let $P_\sigma$ be the associated permutation matrix given by $P_\sigma = (p_{ij})_{i,j=1}^{n-1}$, $p_{ij} = \mathbf{1}_{i=\sigma(j)}$ and $M_\sigma$ be the $n \times n$ matrix obtained by adding an additional zero row and zero column to $P_\sigma$, in the following way :*

$$M_\sigma := \begin{pmatrix} P_\sigma & \mathbf{0}_v \\ \mathbf{0}_h & 0 \end{pmatrix}$$

*where $0_v$ and $0_h$ are zero column and zero row vectors respectively.*
*Then $(M_\sigma, M_\sigma, M_\sigma) \star t_1 = t_1$, i.e. $(M_\sigma, M_\sigma, M_\sigma)$ is in the stabilizer subgroup of $t_1$.*

*Proof.* As in Lemma 7, the action of the permutation matrices is permuting the order of the bases being used to construct $t_1 = \sum_{i=1}^{n-1} e_i \otimes e_i \otimes e_i$. In this case, however, we risk mapping one of the basis vectors $\{e_1, \dots e_{n-1}\}$ to $e_n$, which should not be included in our summation. Hence it is crucial to exclude the permutations that do so by forcing the final row/column to be zeros. This leaves us with the permutations from $S_{n-1}$. □

**Lemma 9.** *Let $I$ be the $(n-1) \times (n-1)$ identity matrix, $\mathbf{v} = (v_1, ..., v_{n-1}) \in \mathbb{F}_q^{n-1}, v_n \in \mathbb{F}_q$ and let $M$ be a matrix of the following form:*

$$M := \begin{pmatrix} I & \mathbf{v} \\ \mathbf{0}_h & v_n \end{pmatrix}$$

*where $0_h$ is a zero row vector. Then $(M, I, I) \star t_1 = t_1$, $(I, M, I) \star t_1 = t_1$, and $(I, I, M) \star t_1 = t_1$. i.e. these three families of isomorphisms are in the stabilizer subgroup of $t_1$.*

*Proof.* Recall, applying an isomorphism $(A, B, C)$ to $t_1$ means

$$(A, B, C) \star t_1 = \sum_{i=1}^{n-1} Ae_i \otimes Be_i \otimes Ce_i.$$

Since we are restricted to $i < n$, every vector contains a zero in the $n^{th}$ component. Thus the $n^{th}$ columns of $A, B, C$ will always be multiplied by zeros. Therefore the families of isomorphisms $\{(M, I, I), (I, M, I), (I, I, M)\}$ are contained in the stabilizer subgroup of $t_1$. □

*Equivalence classes on $G$.* The study of stabilizer groups leads us to the definition of an equivalence relation on $G = \mathrm{GL}(n, q) \times \mathrm{GL}(n, q) \times \mathrm{GL}(n, q)$. This observation will allow us to add more constraints to the solutions we are looking for in CTI (Computational Tensor Isomorphism Problem), and thus accelerate our attack in the proceeding section.

**Definition 12.** *Let $(A_1, B_1, C_1), (A_2, B_2, C_2)$ be two elements from $G$, and $t \in \mathbf{V}$ a tensor. We say that $(A_1, B_1, C_1)$ and $(A_2, B_2, C_2)$ belong to the same* equivalence class via t*, denoted $(A_1, B_1, C_1) \equiv_t (A_2, B_2, C_2)$, if and only if they are equal up to right-multiplication by an element in the stabilizer subgroup of $t$.*

### 3.4 Solving CTI in orbits of $t_0$

In this section we focus on the $b = 0$ case from the commitment scheme, and outline an approach to recovering an element from the isomorphism class via $t_0$ of the secret $(A, B, C)$. Recall our problem is as follows:

*Problem 6.* Let $t_0, g \in \mathbf{V}$ be two 3-tensors such that

$$t_0 := \sum_{i=1}^{n} e_i \otimes e_i \otimes e_i, \text{ and } g := \sum_{i,j,k=1}^{n} g_{i,j,k} e_i \otimes e_j \otimes e_k$$

for $\{g_{i,j,k}\} \subset \mathbb{F}_q$. Assuming it exists, find an isomorphism $(A, B, C) \in G$ such that

$$(A, B, C) \star t_0 = g, \tag{4}$$

where $\star$ is as defined in Equation 3.3.

We begin by noticing that the tensor $t_0$ has exactly $n$ rank 1 points (up to scalars), which we will prove in Lemma 10. Then from Lemma 3 this tells us that $g$ will also have $n$ rank 1 points. Denote a set of rank 1 points of $t_0$ as $\{e_1, \dots e_n\}$. Suppose we could find the rank 1 points in $g$, denoted $\{a_1, \dots a_n\}$. Then, we can try to match the $e_i$ to the $a_i$ to recover part of the isomorphism, namely $A^{-1}$. We formalize this idea in Lemma 11.

**Lemma 10.** *Let $t_0 = \sum_{i=1}^{n} e_i \otimes e_i \otimes e_i$ then $t_0$ has exactly $n$ rank 1 points (up to scalars).*

*Proof.* Let $u := [u_1, \dots u_n]$ be a rank 1 point in $t_0$, define matrices $T_0^1, \dots T_0^n$ as

$$(T_b^k)_{ij} = \begin{cases} 1 \text{ if } i = j = k \text{ and } k \leq n - b \\ 0 \text{ otherwise} \end{cases}.$$

Since $u$ has rank 1 in $t_0$, we expect the following diagonal matrix to have rank 1 (by definition of rank of a point, see Definition 11).

$$U_0 := \begin{pmatrix} u_1 & & 0 \\ & \ddots & \\ 0 & & u_n \end{pmatrix}$$

Thus, for $U_0$ to have rank 1, we see that $u$ can have only one non-zero entry. This gives $n$ distinct points up to scalars. □

**Lemma 11.** *Consider Problem 6. Let $\{a_1, \dots, a_n\}$ be the set of rank 1 points of $g$, and $\{e_1, \dots, e_n\}$ be the set of rank 1 points in $t_0$ (up to scalar multiplication). Then there exists an ordering, $\sigma$, of $\{a_1, ..., a_n\}$ and a matrix $A$ such that for each $i \in [1, \dots n]$, $a_{\sigma(i)} = e_i A^{-1}$.*

*Proof.* As a consequence of Lemma 3, we see that a rank one point in $t_0$ is sent to a rank one point in $g$ through $A^{-1}$. This will fix the ordering $\sigma$.

□

Finding the rank 1 points $\{a_1, \dots a_n\}$ is an instance of MinRank, so we can use the algorithm given in [4] that is summarized in Section 2.5. In this particular context, we want to find $n$ matrices with target rank 1. This means that our coefficient variable matrix, $R$, is simply a row matrix. Denote the entries in $R$ as $r_1, \dots r_n$. This leaves us with a bilinear system in the target variables, $\{x_i\}_{i=1}^n$, and the coefficient variables, $\{r_i\}_{i=1}^n$, where we will have $n^2(n-1)/2$ equations and $n(n-1)$ monomials. If the equations are not related by linear dependencies,

we can solve the system by direct linearization. In our experiments, this was indeed the case.

Going forward, Lemma 11 shows that we can recover some matrix $A_0^{-1}$ whose rows are given by $\{a_i\}_{i=1}^n$. Our findings from Section 3.3 tell us that neither the permutation of these rows, nor the scalar multiples of them, will affect our search for a suitable isomorphism satisfying Equation 4. In fact, fixing these rows and scalar multiples is a good idea as it reduces the number of solutions and variables in our search.

Once we have recovered some $A_0^{-1}$, we can then solve for $B_0$ and $C_0$ to complete the isomorphism by considering the system of $n^3$ linear equations in $2n^2$ variables given by $(I, B_0, I)\star t_0 = (A_0^{-1}, I, C_0^{-1})\star g$. The system, as is, is very under-determined because of the size of the equivalence class of $(A, B, C)$ via $t_0$. To slim down the solution space, thanks to Lemma 6, we can similarly normalize the first row of $B_0$, by setting the entries of this row to arbitrary values of $\mathbb{F}_q^*$.

There is a chance that this will not give a solution, in case one or more of the entries in the first row of $B$ were 0. We argue in the proof of Theorem 3 that when $q$ is large enough compared to $n$, the probability of this failure is very low. In the small chance of failure, however, we can apply a new known isomorphism, $(I, B_1, I) \in G$, and now consider $(I, B_1, I)\star((A, B, C)\star t_0) = (I, B_1, I)\star g$ instead. This essentially "rerandomizes" our instance, in the hopes that the solution space for $B_1 \cdot B$ will contain an entry whose first row has no zeros. Alternatively, we can also normalize a randomly chosen element in each column of $B$.

These steps are summarized in Figure 4.

1: **Given**: $g \in \mathbf{V}$, isomorphic to $t_0$.
2: $a_1, \ldots a_n \leftarrow$ `MinRank`$(g, \text{target rank} = 1)$
3: $A_0^{-1} \leftarrow$ `Matrix`$(a_1, \ldots a_n)$
4: $B_0, C_0 \leftarrow$ `Solutions`$((I, B_0, I)\star t_0 = (A_0^{-1}, I, C_0^{-1})\star g) \cap$ `Solutions`$((B_0)_{1,i} = 1, i = 1, \ldots n)$
5: **return** $(A_0, B_0, C_0)$

**Fig. 4.** Solving Problem 6.

**Theorem 3.** *There exists a probabilistic algorithm that solves Problem 6 using $O(n^6)$ operations.*

*Proof.* We first show that the probability of failure of Algorithm 4 is negligible for large $q$. When $q$ is small, in case of failure one can just apply a new isomorphism $(I, B_1, I) \in G$ to $g$, and repeat the algorithm with this new instance as described above.

We have a chance of failure when we normalize the first row of $B_0$. Since we have already fixed $A_0$, this fixes the permutation of columns of $B_0$ as well (see Lemma 7). Hence, we risk failure when one of these normalized columns necessarily has a leading zero. The chance of this happening is the complement

probability to the columns having no leading zero. This gives us a total probability of failure of

$$P(\text{Alg } 4 \text{ fails}) = 1 - \frac{(q-1)^n}{q^n},$$

which is negligible for large $q$.

We now prove the complexity of this algorithm. The dominating subroutines in Algorithm 4 include the MinRank computation, the tensor group action arithmetic, and solving the final system of equations (which can be done using Gaussian elimination). While the tensor arithmetic could be done in time $O(n^4)$, as noted in Section 6.2 of [48], the other two subroutines require $O(n^6)$ operations. □

*Remark 4.* An alternate approach to proving Theorem 3 could be to apply Lemma 11 over all three axis, allowing us to recover some $A, B, C$, each up to some stabilizer elements. The issue with this approach, however, is that those stabilizer elements will a priori not be the same for all three matrices, so at this point we can write a system of equations to recover these stabilizer elements and finish the attack. This approach, while correct, adds an extra step, making it neither conceptually simpler nor more efficient than what is outlined above. Additionally, the above approach has the advantage of directly using the partial knowledge we have about A when recovering B and C.

We test the attack from Theorem 3 on the parameters from Section 2.4 and give the results in seconds in Table 1. We consider some additional parameters in Figure 5 to further support our complexity claims.

| $n$ | $q$ | time (s) |
|---|---|---|
| 14 | 4093 | 9.3 |
| 22 | 4093 | 141.6 |
| 30 | 2039 | 858.9 |

**Table 1.** Timings in seconds for solving the CTI variant from Problem 6.

### 3.5 Solving CTI in orbits of $t_1$

The approach of computing the set of rank 1 points via the MinRank algorithm from [4] will only work in the case the tensor has rank $n$ (i.e. $b = 0$ in the commitment scheme). Any smaller rank would result in too many rank 1 points to compute. In the $b = 1$ case, however, we will see that we can edit MinRank to further filter distinct rank 1 points using our knowledge of the stabilizer group of $t_1$.

We begin by computing the one rank 0 point in $g$ (which is unique up to scalars). This point is easy to compute, as seen in Section 3.2. Denote it $P_0$.

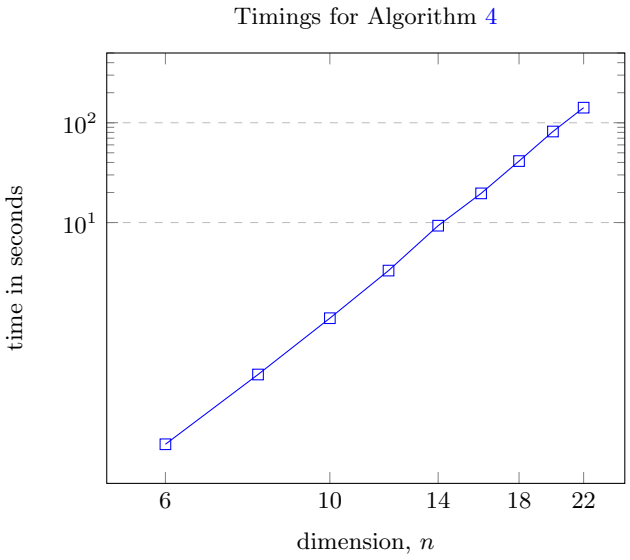


**Fig. 5.** We show some timings of Algorithm 4 for $q = 4093$ and varying values of $n$ on a log-log scale. This corroborates our claim that Algorithm 4 has complexity $O(n^6)$, since the curve follows very closely to a line of slope 6.

Now, we know that any rank 1 point could have a rank 0 point added to it, and still remain a rank 1 point. In other words, we get an equivalence class on the set of rank 1 points where we say two rank 1 points, $R_i, R_j$, are equivalent if $R_i - R_j = \lambda P_0$ for some scalar $\lambda$. This means, for each $i \in [1, \dots n-1]$, the equivalence class can be written as the subspace $\langle P_i, P_0 \rangle$, for some rank 1 point $P_i$, and the rank 0 point $P_0$. Hence, we can obtain a representative from each equivalence class, $P_i$, by appropriately filtering the results returned from MinRank. More precisely, we normalize a component of the rank 1 points, thus filtering much of the equivalence class. The index of the component to be normalized corresponds to the index of the first non-zero entry of $P_0$.

After these computations, we will be left with one rank 0 point and $n-1$ distinct rank 1 points. We are assured to get exactly $n-1$ linearly independent rank 1 points since each equivalence class, $\langle P_i, P_0 \rangle$, is the image under an isomorphism of the subspace $\langle e_i, e_n \rangle$. We can then follow a similar attack as that described in Section 3.4, Figure 4, to reconstruct an isomorphism $(A_0, B_0, C_0) \equiv_{t_1} (A, B, C)$.

First we need to be careful to place the rank 0 point in the final row of $A_0^{-1}$ due to the restriction on permutation matrices in the stabilizer group of $t_1$, as shown in Lemma 8. We can also use Lemma 9 to further filter out options by fixing the final columns of $B_0, C_0$ to random elements. To do so we slightly modify the equations we are solving to avoid high degree equations involving the inverse of $C_0$. We consider the new equation $(I, B_0, C_0) \star t_1 = (A_0^{-1}, I, I) \star g$. This new equation is no longer linear since the left hand side will have quadratic

terms in the variables of $B_0$ and $C_0$. This means we cannot solve the system of equations linearly, but instead will require a Gröbner basis. Experiments show the run times of this attack to be very close to those of the computational attack on $t_0$, which we give in Table 2. Thus, we also estimate the complexity of the overall attack to be polynomial.

| $n$ | $q$ | time (s) |
|---|---|---|
| 14 | 4093 | 8.7 |
| 22 | 4093 | 158.1 |
| 30 | 2039 | 1235.9 |

**Table 2.** Timings in seconds for solving CTI for $t_1$.

## 4 Repairing the Asiacrypt 2023 scheme

In this section, we propose a new construction of a commitment scheme and include proofs of its security. This commitment is the first non-interactive commitment scheme from non-transitive group actions proposed; such group actions are less restrictive, and arise naturally. In particular, tensor-based group actions are such an example. At the moment isogenies have been among the only group actions receiving attention. By studying tensors in this application, and showing how they can be used in protocols, our hope is to contribute to the assumption diversification of the field.

The key components that we aim to preserve from [20] are the following: a commitment scheme based on a non-transitive group action for which the group action should be uncertified[5], and the commitment scheme should be non-interactive.

We therefore propose a non-interactive bit commitment scheme from non-transitive uncertified group actions, described in Figure 6. It is statistically binding (Theorem 4) and computationally hiding (Theorem 5). We improve on [32]'s construction, which is also statistically binding and computationally hiding but requires interaction. We lose the perfect binding property of [20], however, this allows us to have a concrete instantiation that relies on standard problems. In practice, replacing perfect binding by statistical binding does not affect the security of more advanced constructions that could be built from bit commitments. Note that we do not keep the framework of Problem 5, but rather introduce a slightly different and simpler one, which still does not introduce any new assumptions.

[5] An *uncertified* group action is a group action for which checking that two elements are in the same orbit is hard.

**Definition 13 (Decisional group action framework).** *Consider a group action $\star : G \times V \to V$ that has the following properties :*

1. *It is an Effective Group Action.*
2. *Given two elements $v_0, v_1$ such that $v_1 = g \star v_0$ for some $g \in G$, computing $g$ is hard.*
3. *The probability, $p$, that two random elements are in different orbits is $p = 1 - negl(\lambda)$.*
4. *The Decisional Group Action Inversion Problem for $\star$ is hard (recall Definition 4).*

Note that the first three properties are described in [32].

$\underline{\mathsf{Setup}(1^\lambda)}$
1: $v_0, v_1 \xleftarrow{\$} V$
2: **return** $v_0, v_1$

$\underline{\mathsf{Commit}(b)}$
1: $g \xleftarrow{\$} G$
2: $c \leftarrow g \star v_b$
3: **return** $c$

$\underline{\mathsf{Verify}(c, (b, g))}$
1: **return** $(c = g \star v_b)$

**Fig. 6.** Commitment scheme

We can instantiate this framework using tensors through the group action described in Equation 3.3. The crucial difference with the construction of [20] is that $v_0, v_1$ are now two **randomly** generated tensors and they do not possess any special structure. In particular, a random tensor will only have low rank points with negligible probability, as shown in the Lemma 12.

*Remark 5.* One can extend the commitment scheme to accommodate a polynomial number of messages say $0, \dots N$ by slightly modifying the Setup and Commit algorithms of Figure 6. Indeed, during Setup one can instead randomly sample $N$ elements $v_0, \dots v_N$ and output these. Then during Commit, to commit to a message $k \in \{0, \dots N\}$, one samples $g \in G$ and computes $g \star v_k$.

*Remark 6.* Note that $v_0, v_1$ must be sampled randomly, in practice this can be done by sampling them as the output of a pseudo-random generator or a cryptographic hash function. What matters for our proofs is that they are indeed indistinguishable from elements sampled from a uniformly random distribution.

**Lemma 12.** *Given a random tensor $t \in \mathbf{V}$, the average number of rank $n - d$ points (up to scalar multiplication) is $q^{-d^2+n-1}$ as $q$ goes to infinity.*

*Proof.* Consider our tensor $t$ as a list of $n$ random $n \times n$ matrices, over $\mathbb{F}_q$, say $M_1, \dots M_n$. A rank $r$ point in $t$ is $\lambda = (\lambda_1, \dots, \lambda_n) \in \mathbb{F}_q^n$, $\lambda \neq 0$, such that $rank(\lambda_1 M_1 + \dots + \lambda_n M_n) = r$.
First, we have that the probability that a random matrix $M$ has rank $n - d$ for some integer $d$ goes to $q^{-d^2}$ as $q$ goes to infinity. This result is a direct observation from [25]. Then we observe that there are about $q^{n-1}$ possibilities projectively for $\lambda$, hence we get that the average number of rank $n - d$ points is $q^{-d^2+n-1}$. □

This means that our distinguishing attack from Section 3.2 is not applicable on a randomly generated instance. Indeed, this lemma shows that it is unlikely for a random tensor to have points of small rank. For points of larger rank, the MinRank attack described in Section 2.5 becomes impractical.

Let us now check that all the desired properties from Definition 13 are satisfied: Property 1 has been studied in [32] and Property 2 corresponds to the Computational Tensor Isomorphism Problem (Problem 3). Property 4 is introduced in [20] and they discuss its applicability to tensors. While we show it is broken for their instantiation using unit tensors, $t_b$, it is still believed to be hard in the case of random tensors, which is precisely our case. Regarding Property 3, we must compute the probability, $p$, of two random tensors being in different orbits under $\star$. Note that [32] assumes that this probability is high but they do not give any estimates.

**Lemma 13.** *The probability, $p$, of two random tensors being in distinct orbits under $\star$ is $p \geq 1 - \frac{1}{q^{n^3-3n^2}}$.*

*Proof.* Let $t \in \mathbf{V}$ be any tensor, then trivially

$$\#Orb(t) \leq \#G = (\#\,\mathrm{GL}_n(q))^3 = O(q^{3n^2}).$$

Furthermore, we have $\#\mathbf{V} = q^{n^3}$. So the total number of orbits is at least $\#\mathbf{V}/(\#\,\mathrm{GL}_n(q))^3 = O(q^{n^3-3n^2})$. The probability of two tensors being in the same orbit is therefore bounded by $\frac{1}{q^{n^3-3n^2}}$, hence $p \geq 1 - \frac{1}{q^{n^3-3n^2}}$. □

We have shown that all the properties from Definition 13 are satisfied. Now it remains to show that our proposed commitment scheme from Figure 6 is both hiding and binding.

**Theorem 4 (Binding).** *The commitment scheme described in Figure 6 is statistically binding.*

*Proof.* Consider $\mathsf{G}$ to be the binding security game from Section 2.3 applied to our scheme. Taking the following probabilities over the randomness of the game we have :

$$\begin{aligned}\Pr[\mathsf{G}(\mathcal{A}) \to 1] &= \Pr[\mathsf{G}(\mathcal{A}) \to 1 \wedge v_0, v_1 \text{ are in the same orbit}] \\ &\quad + \Pr[\mathsf{G}(\mathcal{A}) \to 1 \wedge v_0, v_1 \text{ are in different orbits}] \\ &\leq \Pr[v_0, v_1 \text{ are in the same orbit}] \\ &\quad + \Pr[\mathsf{G}(\mathcal{A}) \to 1 | v_0, v_1 \text{ are in different orbits}] \cdot p,\end{aligned}$$

where $p$ is the probability defined by Property 3, and computed in Lemma 13. We have $\Pr[v_0, v_1 \text{ are in the same orbit}] = 1 - p$ by definition of $p$. Furthermore, $\Pr[\mathsf{G}(\mathcal{A}) \to 1 | v_0, v_1 \text{ are in different orbits}] = 0$ since otherwise if $\mathcal{A}$ wins $\mathsf{G}$ in that case, this means they return $(A_0, B_0, C_0)$ and $(A_1, B_1, C_1)$ such that

$$(A_0, B_0, C_0) \star v_0 = (A_1, B_1, C_1) \star v_1.$$

In particular this would mean that $(A_1^{-1}A_0, B_1^{-1}B_0, C_1^{-1}C_0) \star v_0 = v_1$ which is impossible since $v_0, v_1$ are in different orbits and thus no such isomorphism exists.

Overall we get

$$\Pr[\mathsf{G}(\mathcal{A}) \to 1] \leq (1-p) + 0 \cdot p = (1-p) = \mathrm{negl}(\lambda).$$

□

*Remark 7.* Notice that in the case where $v_0, v_1$ are indeed in the same orbit (which happens only with negligible probability), breaking the Binding problem amounts to solving the Computational Tensor Isomorphism problem (Problem 3) in a *random* orbit, hence remains plausibly hard.

**Theorem 5 (Hiding).** *Assuming the hardness of the dGA-IP problem (Definition 4), our commitment scheme is hiding.*

*Proof.* Consider $\mathsf{G}$ to be the hiding game from Section 2.3. Taking the following probabilities over the randomness of the game, we have :

$$\begin{aligned}\Pr[\mathsf{G}(\mathcal{A}) \to 1] &= \Pr[\mathsf{G}(\mathcal{A}) \to 1 \wedge v_0, v_1 \text{ are in the same orbit}] \\ &\quad + \Pr[\mathsf{G}(\mathcal{A}) \to 1 \wedge v_0, v_1 \text{ are in different orbits}] \\ &\leq \Pr[v_0, v_1 \text{ are in the same orbit}] \\ &\quad + \Pr[\mathsf{G}(\mathcal{A}) \to 1 | v_0, v_1 \text{ are in different orbits}] \cdot p,\end{aligned}$$

where $p$ is the probability defined by Property 3. We have $\Pr[v_0, v_1 \text{ are in the same orbit}] = 1 - p = \mathrm{negl}(\lambda)$.
In the case where $v_0, v_1$ are in different orbits, one can refer to [20], Theorem 2. Roughly, given an adversary $\mathcal{B}$ against dGA-IP, that receives as input $s, t$, they instantiate two Hiding games with adversaries $\mathcal{A}_1, \mathcal{A}_2$ to which they respectively give $s$ or $t$ as input, and $\mathcal{B}$ returns 1 if the outputs of $\mathcal{A}_1$ and $\mathcal{A}_2$ are equal and 0 otherwise. They get that

$$\Pr[\mathsf{G}(\mathcal{A}) \to 1 | v_0, v_1 \text{ are in different orbits}] \leq \frac{1}{2} + 2\epsilon(\lambda)^2$$

where $\epsilon(\lambda)$ is the advantage in the dGA-IP game.

So overall we have

$$\begin{aligned}\Pr[\mathsf{G}(\mathcal{A}) \to 1] &\leq (1-p) + p \cdot (\frac{1}{2} + 2\epsilon(\lambda)^2) \\ &= \frac{1}{2}p + (1-p) + 2p\epsilon(\lambda)^2\end{aligned}$$

and by assumption both $\epsilon(\lambda)$ and $1-p$ are negligible. □

*Remark 8.* Consider the extension of our commitment scheme to a message space of $\{0, \dots N\}$ as described in Remark 5. Then the hiding and binding properties are derived in a straightforward way since $v_0, \dots v_N$ are in different orbits with probability at least $1 - \frac{N}{q^{n^3-3n^2}}$.

### 4.1 Cost comparison

The cost of generating the commitment in [20] is $O(n^3)$ multiplications and additions over $\mathbb{F}_q$ (the cost of matrix multiplication, ignoring asymptotic improvements). Due to our use of generic tensors, we have to perform $3n$ matrix multiplications to compute our commitment, bringing the cost to $O(n^4)$. The cost of multiplications over $\mathbb{F}_q$ is $O(\log q \log\log q)$. While this provides a slowdown compared to [20], it is still reasonable when compared to other cryptographic operations. In particular, we benefit from the fact that matrix multiplication already has existing low-level code and algorithms, including parallel algorithms. It is also worth noting that our loss in efficiency only occurs during the actual computation of the commitment. For any operation on the commitment itself or involving it (e.g zero-knowledge proofs) the cost of manipulating it would be the same for both our construction and that of [20] since they are both comprised of matrix multiplication on a (seemingly) random tensor.

MEDS [16], a NIST signature candidate, only uses two matrices $A, B$, but introduces *systematic form computation* to recover a third matrix, $C$. This allows a trade-off between computation time and size. Therefore, the cost of a single group action on a matrix code is the cost of matrix multiplication plus the systematic form computation. This trade-off could also be applied to the case of our commitment scheme, along with their other (future) group action optimizations.

### 4.2 On proofs of knowledge

To use commitment schemes within more advanced protocols such as verifiable secret sharing or zero-knowledge proofs for generic statements, it is useful to have non-interactive zero-knowledge proofs of knowledge of commitments. More precisely, one wishes to prove that they know the value to which a commitment $c$ corresponds without giving away any information about it.

Let us consider the commitment scheme of Figure 6, extended to the message space $\{0, \dots N\}$. Given public parameters $\mathsf{pp} = \{t_0, \dots t_N\}$ consisting of $N+1$ random tensors from $\mathbf{V}$, we consider the relation

$$\mathcal{R} = \{(c, (i, (A, B, C))) \mid c = (A, B, C) \star t_i\},$$

where $c$ is the statement and $(i, (A, B, C))$ is the witness. This means given $c$ we wish to prove knowledge of the message $i$ and randomness $(A, B, C)$ to which $c$ corresponds, without revealing anything about them. We briefly expose two ways of obtaining such a proof.

**Using a variant of OR-proofs.** A proof for $\mathcal{R}$ can be obtained by combining a variant of an OR-proof [19] and a proof for $\mathcal{R}_i = \{((c, t_i), (A, B, C))) \mid c = (A, B, C) \star t_i\}$, to show that one knows an isomorphism mapping $c$ to a particular choice of one of the $t_i$. A proof for $\mathcal{R}$ can be obtained by adapting, for example, the graph isomorphism proof [26]. However, using OR-proofs increases the communication cost in the answer of the prover, and we will therefore prefer to build a direct proof for the full relation $\mathcal{R}$.

**A direct proof for $\mathcal{R}$.** We give in Figure 7 a sigma protocol that directly gives a proof for $\mathcal{R}$.

$\underline{\mathsf{Prover}_1(\mathsf{pp}, i, c, (A, B, C))}$

1: $A', B', C' \xleftarrow{\$} \mathrm{GL}_n(q)^3$
2: $\sigma \xleftarrow{\$} S_n$
3: **for** $j = 1, \ldots N$ **do**
4: $\quad d_j \leftarrow (A', B', C') \star t_{\sigma(j)}$
5: **end for**
6: $st \leftarrow (\mathsf{pp}, (A, B, C), (A', B', C'), \sigma)$
7: **Send** $(d_j)_{j=1}^N$ to Verifier
8: **return** $st$

$\underline{\mathsf{Prover}_2(st, \mathsf{ch})}$

1: **if** $\mathsf{ch} = 0$ **then**
2: $\quad \mathsf{resp} \leftarrow ((A', B', C'), \sigma)$
3: $\quad$ **return** resp
4: **end if**
5: $\mathsf{resp} \leftarrow (A(A')^{-1}, B(B')^{-1}, C(C')^{-1}), \sigma(i))$
6: **return** resp

$\underline{\mathsf{Verifier}_1(\mathsf{pp}, c, (d_j)_{j=1}^N)}$

1: $\mathsf{ch} \xleftarrow{\$} \{0, 1\}$
2: $st \leftarrow (\mathsf{pp}, \mathsf{ch}, c, (d_j)_{j=1}^N), t_i)$
3: **Send** ch to Prover
4: **return** $st$

$\underline{\mathsf{Verifier}_2(st, \mathsf{resp})}$

1: **if** $\mathsf{ch} = 0$ **then**
2: $\quad \mathsf{resp} \to ((\tilde{A}, \tilde{B}, \tilde{C}), \sigma)$
3: $\quad$ **return** $\forall i, (\tilde{A}, \tilde{B}, \tilde{C}) \star t_{\sigma(i)} = d_i\big)$
4: **end if**
5: $\mathsf{resp} \to (\tilde{A}, \tilde{B}, \tilde{C}), i')$
6: **return** $\big((\tilde{A}, \tilde{B}, \tilde{C}) \star d_{i'} = c\big)$

**Fig. 7.** Proof system for $\mathcal{R}$

We sketch the extractor for special soundness and simulator for honest-verifier zero-knowledge below.

*Special-soundness.* Suppose we have two accepting transcripts with different challenges, $((d_j)_{j=1}^N, \mathsf{ch}_0, \mathsf{resp}_0)$ and $((d_j)_{j=1}^N, \mathsf{ch}_1, \mathsf{resp}_1)$. Without loss of generality, we assume $\mathsf{ch}_0 = 0$. Then we can build an extractor $\mathcal{E}$ which, given these transcripts as input, does the following : Recover $(A', B', C')$ from $\mathsf{resp}_0$ and $(A(A')^{-1}, B(B')^{-1}, C(C')^{-1})$ from $\mathsf{resp}_1$. Compute $(A, B, C)$ by multiplying the latter by the former. Recover $i'$ from $\mathsf{resp}_0$ and recover the correct $i = \sigma^{-1}(i')$. Then $(i, (A, B, C))$ is the witness.

*Honest-verifier zero-knowledge.* We now sketch the simulator. In the case $\mathsf{ch} = 0$, the simulator just follows the protocol honestly and will be producing an accepting transcript. In the $\mathsf{ch} = 1$ case, the simulator samples $(A', B', C')$ and $\sigma$ as in the protocol, then picks some index $j$ and sets $d_j = ((A')^{-1}, (B')^{-1}, (C')^{-1}) \star c$. The other $d_i$ for $i \neq j$ are computed according to the protocol. The distribution will remain computationally indistinguishable since $(A', B', C')$ is random. The answer then consists of $j$ and $(A', B', C')$.

## 5 Conclusion

In this work, we study low rank points and stabilizers on unit tensors. We show how this information gives us the tools to break the hiding property of the bit

commitment scheme proposed in [20] at Asiacrypt 2023, that uses these unit tensors. Our attacks further allow to recover all the secret information used during the creation of the commitment. All of these algorithms run in polynomial time, which leaves the framework introduced in [20] with no concrete instantiation. Prior to this work, there was no evidence in the literature suggesting that unit tensors were not secure for use in cryptography.

With these attacks in mind, we propose a slightly different framework as well as a construction that makes use of *random* tensors. This allows us to build a statistically binding, computationally hiding commitment from non-transitive group actions. Finally, we complete this work by proposing a zero-knowledge proof of opening for our new commitment scheme.

At the moment, as pointed out by [20], the field of commitment schemes from non-transitive group actions is lacking protocols that are both post-quantum and non-interactive. We hope our new construction helps to fill this gap, and that our attacks can help to inform future cryptography designers on some of the limits of working with unit tensors. We leave any further investigation into a commitment scheme using the framework proposed in [20] as future work.

## Supplementary Material

## A Proofs

The proofs included here are done via a very computational approach, using core definitions. We encourage the reader who wants to familiarize themselves with these definitions to do them as exercise.

**Lemma 5.** *Let $\lambda_a, \lambda_b, \lambda_c \in \mathbb{F}_q$ be such that $\lambda_a\lambda_b\lambda_c = 1$. Then for all $v \in \mathbf{V}$, we have that $(\lambda_a I_n, \lambda_b I_n, \lambda_c I_n) \star v = v$.*

*Proof.* Let us write $v = \sum_{i,j,k=1}^{n} v(i,j,k) e_i \otimes e_j \otimes e_k$. We have

$$\begin{aligned}
(\lambda_a I_n, \lambda_b I_n, \lambda_c I_n) \star v &= \sum_{i,j,k=1}^{n} v(i,j,k)(\lambda_a I_n)e_i \otimes (\lambda_b I_n)e_j \otimes (\lambda_c I_n)e_k \\
&= \sum_{i,j,k=1}^{n} v(i,j,k)\lambda_a\lambda_b\lambda_c e_i \otimes e_j \otimes e_k \\
&= \sum_{i,j,k=1}^{n} v(i,j,k) e_i \otimes e_j \otimes e_k \text{ since } \lambda_a\lambda_b\lambda_c = 1 \\
&= v
\end{aligned}$$

□

**Lemma 7.** *Let $\sigma \in S_n$, then there exists $P_\sigma$ such that $(P_\sigma, P_\sigma, P_\sigma) \star t_0 = t_0$, given by $P_\sigma = (p_{ij})_{i,j=1}^n$ and $p_{ij} = \mathbf{1}_{i=\sigma(j)}$.*

*Proof.* Let $t_0 = \sum_{s=1}^{n} e_s \otimes e_s \otimes e_s$, then we can express the action $(A,B,C) \star t_0$ (from (3.3)) as $(A,B,C) * (\sum_{s=1}^{n} e_s \otimes e_s \otimes e_s) = \sum_{i,j,k}^{n} \sum_{s=1}^{n} A_{is}B_{js}C_{ks} e_i \otimes e_j \otimes e_k$.

$$\begin{aligned}
(P_\sigma, P_\sigma, P_\sigma) \star t_0 &= \sum_{i,j,k=1}^{n} \sum_{s=1}^{n} (P_\sigma)_{is}(P_\sigma)_{js}(P_\sigma)_{ks} e_i \otimes e_j \otimes e_k \\
&= \sum_{i,j,k=1}^{n} \sum_{s=1}^{n} \mathbf{1}_{i=\sigma(s)}\mathbf{1}_{j=\sigma(s)}\mathbf{1}_{k=\sigma(s)} e_i \otimes e_j \otimes e_k \\
&= \sum_{i,j,k=1}^{n} \sum_{s=1}^{n} \mathbf{1}_{i=j=k}\mathbf{1}_{i=\sigma(s)} e_i \otimes e_j \otimes e_k \\
&= \sum_{m=1}^{n} \sum_{s=1}^{n} \mathbf{1}_{m=\sigma(s)} e_m \otimes e_m \otimes e_m \\
&= \sum_{m=1}^{n} e_m \otimes e_m \otimes e_m = t_0
\end{aligned}$$

since $\sigma$ is a bijection. □

## References


1. Alamati, N., Feo, L.D., Montgomery, H., Patranabis, S.: Cryptographic group actions and applications. In: Moriai, S., Wang, H. (eds.) 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. Lecture Notes in Computer Science, vol. 12492, pp. 411–439. Springer (2020)
2. Albrecht, M.R., Fenzi, G., Lapiha, O., Nguyen, N.K.: SLAP: succinct lattice-based polynomial commitments from standard assumptions. Cryptology ePrint Archive, Paper 2023/1469 (2023), https://eprint.iacr.org/2023/1469
3. Aragon, N., Gaborit, P., Hauteville, A., Tillich, J.: A new algorithm for solving the rank syndrome decoding problem. In: 2018 IEEE International Symposium on Information Theory, ISIT 2018, Vail, CO, USA, June 17-22, 2018. pp. 2421–2425. IEEE (2018). https://doi.org/10.1109/ISIT.2018.8437464
4. Bardet, M., Bros, M., Cabarcas, D., Gaborit, P., Perlner, R.A., Smith-Tone, D., Tillich, J., Verbel, J.A.: Improvements of algebraic attacks for solving the rank decoding and MinRank problems. In: Moriai, S., Wang, H. (eds.) 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. Lecture Notes in Computer Science, vol. 12491, pp. 507–536. Springer (2020). https://doi.org/10.1007/978-3-030-64837-4_17
5. Barenghi, A., Biasse, J., Persichetti, E., Santini, P.: On the computational hardness of the code equivalence problem in cryptography. Adv. Math. Commun. **17**(1), 23–55 (2023). https://doi.org/10.3934/AMC.2022064
6. Baum, C., Damgård, I., Lyubashevsky, V., Oechsner, S., Peikert, C.: More efficient commitments from structured lattice assumptions. In: Catalano, D., De Prisco, R. (eds.) Security and Cryptography for Networks. pp. 368–385. Springer International Publishing, Cham (2018)
7. Beullens, W.: Not enough LESS: An improved algorithm for solving code equivalence problems over Fq. In: International Conference on Selected Areas in Cryptography. pp. 387–403. Springer (2020)
8. Beullens, W., Kleinjung, T., Vercauteren, F.: CSI-FiSh: Efficient isogeny based signatures through class group computations. Advances in Cryptology – ASIACRYPT 2019 pp. 227–247 (2019)
9. Bläser, M., Duong, D.H., Narayanan, A.K., Plantard, T., Qiao, Y., Sipasseuth, A., Tang, G.: The alteq signature scheme: Algorithm specifications and supporting documentation (2023), https://pqcalteq.github.io/ALTEQ_spec_2024.03.05.pdf
10. Bouillaguet, C., Fouque, P., Véber, A.: Graph-theoretic algorithms for the "isomorphism of polynomials" problem. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7881, pp. 211–227. Springer (2013). https://doi.org/10.1007/978-3-642-38348-9_13, https://doi.org/10.1007/978-3-642-38348-9_13
11. Brassard, G., Yung, M.: One-way group actions. In: Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology. p. 94–107. CRYPTO '90, Springer-Verlag, Berlin, Heidelberg (1990)
12. Bürgisser, P., Ikenmeyer, C.: Geometric complexity theory and tensor rank. CoRR **abs/1011.1350** (2010), http://arxiv.org/abs/1011.1350

13. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. In: Peyrin, T., Galbraith, S.D. (eds.) Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III. Lecture Notes in Computer Science, vol. 11274, pp. 395–427. Springer (2018). https://doi.org/10.1007/978-3-030-03332-3_15
14. Chen, M., Lai, Y., Laval, A., Marco, L., Petit, C.: Malleable commitments from group actions and zero-knowledge proofs for circuits based on isogenies. In: Chattopadhyay, A., Bhasin, S., Picek, S., Rebeiro, C. (eds.) Progress in Cryptology - INDOCRYPT 2023 - 24th International Conference on Cryptology in India, Goa, India, December 10-13, 2023, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14459, pp. 221–243. Springer (2023). https://doi.org/10.1007/978-3-031-56232-7_11, https://doi.org/10.1007/978-3-031-56232-7_11
15. Chou, T., Niederhagen, R., Persichetti, E., Ran, L., Randrianarisoa, T.H., Reijnders, K., Samardjiska, S., Trimoska, M.: Matrix equivalence digital signature (2023), https://www.meds-pqc.org/spec/MEDS-2023-07-26.pdf
16. Chou, T., Niederhagen, R., Persichetti, E., Randrianarisoa, T.H., Reijnders, K., Samardjiska, S., Trimoska, M.: Take your MEDS: digital signatures from matrix code equivalence. In: Mrabet, N.E., Feo, L.D., Duquesne, S. (eds.) Progress in Cryptology - AFRICACRYPT 2023 - 14th International Conference on Cryptology in Africa, Sousse, Tunisia, July 19-21, 2023, Proceedings. Lecture Notes in Computer Science, vol. 14064, pp. 28–52. Springer (2023). https://doi.org/10.1007/978-3-031-37679-5_2
17. Couveignes, J.M.: Hard homogeneous spaces. Cryptology ePrint Archive, Paper 2006/291 (2006), https://eprint.iacr.org/2006/291
18. Couvreur, A., Debris-Alazard, T., Gaborit, P.: On the hardness of code equivalence problems in rank metric. CoRR **abs/2011.04611** (2020), https://arxiv.org/abs/2011.04611
19. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y.G. (ed.) Advances in Cryptology — CRYPTO '94. pp. 174–187. Springer Berlin Heidelberg, Berlin, Heidelberg (1994)
20. D'Alconzo, G., Flamini, A., Gangemi, A.: Non-interactive commitment from non-transitive group actions. Advances in Cryptology – ASIACRYPT 2023 p. 723 (2023)
21. Ducas, L., van Woerden, W.P.J.: On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In: Dunkelman, O., Dziembowski, S. (eds.) 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. Lecture Notes in Computer Science, vol. 13277, pp. 643–673. Springer (2022). https://doi.org/10.1007/978-3-031-07082-2_23
22. Fenzi, G., Moghaddas, H., Nguyen, N.K.: Lattice-based polynomial commitments: Towards asymptotic and concrete efficiency. Cryptology ePrint Archive, Paper 2023/846 (2023), https://eprint.iacr.org/2023/846
23. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings. Lecture Notes in Com-

puter Science, vol. 263, pp. 186–194. Springer (1986). https://doi.org/10.1007/3-540-47721-7_12, https://doi.org/10.1007/3-540-47721-7_12
24. Frederiksen, T.K., Pinkas, B., Yanai, A.: Committed MPC - maliciously secure multiparty computation from homomorphic commitments. In: Abdalla, M., Dahab, R. (eds.) Public-Key Cryptography - PKC 2018 - 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10769, pp. 587–619. Springer (2018). https://doi.org/10.1007/978-3-319-76578-5_20
25. Fulman, J., Goldstein, L.: Stein's method and the rank distribution of random matrices over finite fields. The Annals of Probability **43**(3) (May 2015). https://doi.org/10.1214/13-aop889
26. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. J. ACM **38**(3), 690–728 (jul 1991). https://doi.org/10.1145/116825.116852
27. Grochow, J.A., Qiao, Y.: On the complexity of isomorphism problems for tensors, groups, and polynomials I: tensor isomorphism-completeness. In: Lee, J.R. (ed.) 12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference. LIPIcs, vol. 185, pp. 31:1–31:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPICS.ITCS.2021.31, https://doi.org/10.4230/LIPIcs.ITCS.2021.31
28. Grochow, J.A., Qiao, Y., Tang, G.: Average-case algorithms for testing isomorphism of polynomials, algebras, and multilinear forms. In: Bläser, M., Monmege, B. (eds.) 38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference). LIPIcs, vol. 187, pp. 38:1–38:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPICS.STACS.2021.38, https://doi.org/10.4230/LIPIcs.STACS.2021.38
29. Håstad, J.: Tensor rank is NP-complete. J. Algorithms **11**(4), 644–654 (1990). https://doi.org/10.1016/0196-6774(90)90014-6
30. Hillar, C.J., Lim, L.: Most tensor problems are NP-hard. J. ACM **60**(6), 45:1–45:39 (2013). https://doi.org/10.1145/2512329
31. Jain, A., Krenn, S., Pietrzak, K., Tentes, A.: Commitments and efficient zero-knowledge proofs from learning parity with noise. In: Wang, X., Sako, K. (eds.) Advances in Cryptology – ASIACRYPT 2012. pp. 663–680. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
32. Ji, Z., Qiao, Y., Song, F., Yun, A.: General linear group action on tensors: A candidate for post-quantum cryptography. In: Hofheinz, D., Rosen, A. (eds.) Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11891, pp. 251–281. Springer (2019). https://doi.org/10.1007/978-3-030-36030-6_11
33. Juels, A., Luby, M., Ostrovsky, R.: Security of blind digital signatures (extended abstract). In: Jr., B.S.K. (ed.) Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings. Lecture Notes in Computer Science, vol. 1294, pp. 150–164. Springer (1997). https://doi.org/10.1007/BFB0052233
34. Kipnis, A., Shamir, A.: Cryptanalysis of the oil & vinegar signature scheme. In: Krawczyk, H. (ed.) Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1462, pp. 257–266.

Springer (1998). https://doi.org/10.1007/BFB0055733, https://doi.org/10.1007/BFb0055733
35. Leon, J.: Computing automorphism groups of error-correcting codes. IEEE Transactions on Information Theory **28**(3), 496–511 (1982)
36. Lyubashevsky, V., Nguyen, N.K., Seiler, G.: Shorter lattice-based zero-knowledge proofs via one-time commitments. In: Garay, J.A. (ed.) Public-Key Cryptography - PKC 2021 - 24th IACR International Conference on Practice and Theory of Public Key Cryptography, Virtual Event, May 10-13, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12710, pp. 215–241. Springer (2021). https://doi.org/10.1007/978-3-030-75245-3_9
37. Morozov, K., Roy, P.S., Sakurai, K.: On unconditionally binding code-based commitment schemes. In: Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication. IMCOM '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3022227.3022327, https://doi.org/10.1145/3022227.3022327
38. Naor, M.: Bit commitment using pseudorandomness. J. Cryptol. **4**(2), 151–158 (1991). https://doi.org/10.1007/BF00196774
39. Narayanan, A.K., Qiao, Y., Tang, G.: Algorithms for matrix code and alternating trilinear form equivalences via new isomorphism invariants. In: Joye, M., Leander, G. (eds.) Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part III. Lecture Notes in Computer Science, vol. 14653, pp. 160–187. Springer (2024). https://doi.org/10.1007/978-3-031-58734-4_6, https://doi.org/10.1007/978-3-031-58734-4_6
40. Ostrovsky, R., Persiano, G., Visconti, I.: Simulation-based concurrent non-malleable commitments and decommitments. In: Reingold, O. (ed.) Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5444, pp. 91–108. Springer (2009). https://doi.org/10.1007/978-3-642-00457-5_7
41. Patarin, J.: Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In: Maurer, U.M. (ed.) Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding. Lecture Notes in Computer Science, vol. 1070, pp. 33–48. Springer (1996). https://doi.org/10.1007/3-540-68339-9_4
42. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings. Lecture Notes in Computer Science, vol. 576, pp. 129–140. Springer (1991). https://doi.org/10.1007/3-540-46766-1_9
43. Poelstra, A., Back, A., Friedenbach, M., Maxwell, G., Wuille, P.: Confidential assets. In: Zohar, A., Eyal, I., Teague, V., Clark, J., Bracciali, A., Pintore, F., Sala, M. (eds.) Financial Cryptography and Data Security - FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, March 2, 2018, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10958, pp. 43–63. Springer (2018). https://doi.org/10.1007/978-3-662-58820-8_4
44. Reijnders, K., Samardjiska, S., Trimoska, M.: Hardness estimates of the code equivalence problem in the rank metric. IACR Cryptol. ePrint Arch. p. 276 (2022), https://eprint.iacr.org/2022/276

45. Rostovtsev, A., Stolbunov, A.: Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Paper 2006/145 (2006), https://eprint.iacr.org/2006/145
46. Schaefer, M., Stefankovic, D.: The complexity of tensor rank. Theory Comput. Syst. **62**(5), 1161–1174 (2018). https://doi.org/10.1007/S00224-017-9800-Y
47. Sterner, B.: Commitment schemes from supersingular elliptic curve isogeny graphs. Journal of Mathematical Cryptology (2021)
48. Tang, G., Duong, D.H., Joux, A., Plantard, T., Qiao, Y., Susilo, W.: Practical post-quantum signature schemes from isomorphism problems of trilinear forms. In: Dunkelman, O., Dziembowski, S. (eds.) 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. Lecture Notes in Computer Science, vol. 13277, pp. 582–612. Springer (2022). https://doi.org/10.1007/978-3-031-07082-2_21