# General Linear Group Action on Tensors: A Candidate for Post-Quantum Cryptography

Zhengfeng Ji * Youming Qiao † Fang Song ‡ Aaram Yun §

**Abstract**

Starting from the one-way group action framework of Brassard and Yung (Crypto '90), we revisit building cryptography based on group actions. Several previous candidates for one-way group actions no longer stand, due to progress both on classical algorithms (e.g., graph isomorphism) and quantum algorithms (e.g., discrete logarithm).

We propose the *general linear group action on tensors* as a new candidate to build cryptography based on group actions. Recent works (Futorny–Grochow–Sergeichuk *Lin. Alg. Appl.*, 2019) suggest that the underlying algorithmic problem, the *tensor isomorphism problem*, is the hardest one among several isomorphism testing problems arising from areas including coding theory, computational group theory, and multivariate cryptography. We present evidence to justify the viability of this proposal from comprehensive study of the state-of-art heuristic algorithms, theoretical algorithms and hardness results, as well as quantum algorithms.

We then introduce a new notion called *pseudorandom group actions* to further develop group-action based cryptography. Briefly speaking, given a group $G$ acting on a set $S$, we assume that it is hard to distinguish two distributions of $(s, t)$ either uniformly chosen from $S \times S$, or where $s$ is randomly chosen from $S$ and $t$ is the result of applying a random group action of $g \in G$ on $s$. This subsumes the classical decisional Diffie-Hellman assumption when specialized to a particular group action. We carefully analyze various attack strategies that support the general linear group action on tensors as a candidate for this assumption.

Finally, we establish the quantum security of several cryptographic primitives based on the one-way group action assumption and the pseudorandom group action assumption.

*Centre for Quantum Software and Information, School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, Australia. Zhengfeng.Ji@uts.edu.au

†Centre for Quantum Software and Information, School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, Australia. Youming.Qiao@uts.edu.au

‡Department of Computer Science and Engineering, Texas A&M University, Texas, USA. fang.song@tamu.edu

§Department of Cyber Security, Division of Software Science and Engineering, Ewha Womans University, Seoul, Korea. aaramyun@ewha.ac.kr

# 1 Introduction

Modern cryptography has thrived thanks to the paradigm shift to a formal approach: precise *definition* of security and mathematically sound *proof* of security of a given construction based on accurate *assumptions*. Most notably, computational assumptions originated from specific algebraic problem such as factoring and discrete logarithm have enabled widely deployed cryptosystems.

Clearly, it is imperative to base cryptography on diverse problems to reduce the risk that some problems turn out to be easy. One such effort was by Brassard and Yung soon after the early development of modern cryptography [BY90]. They proposed an approach to use a *group action* to construct a *one-way function*, from which they constructed cryptographic primitives such as bit commitment, identification and digital signature. The abstraction of one-way group actions (OWA) not only unifies the assumptions from factoring and discrete logarithm, but more importantly Brassard and Yung suggested new problems to instantiate it such as the graph isomorphism problem ($\mathsf{GI}$). Since then, many developments fall in this framework [Pat96, Cou06, HS07, MRV07]. In particular, the work of Couveignes [Cou06] can be understood as a specific group action based on isogenies between elliptic curves, and it has spurred the development of *isogeny-based* cryptography [DFJP14].

However, searching for concrete group actions to support this approach turns out to be a tricky task, especially given the potential threats from attackers capable of quantum computation. For graph isomorphism, there are effective heuristic solvers [McK80, MP14] as well as efficient *average-case* algorithms [BES80], not to mention Babai's recent breakthrough of a *quasipolynomial*-time algorithm [Bab16]. Shor's celebrated work solves discrete logarithm and factoring in polynomial time on a *quantum* computer [Sho94], which would break a vast majority of public-key cryptography. The core technique, *quantum Fourier sampling*, has proven powerful and can be applied to break popular symmetric-key cryptosystems as well [KLLNP16]. A *subexponential-time* quantum algorithm was also found for computing isogenies in ordinary curves [CJS14], which attributes to the shift to *super-singular* curves in the recent development of isogeny-based cryptography [GV18]. In fact, there is a considerable effort developing *post-quantum* cryptography that can resist quantum attacks. Besides isogeny-based, there are popular proposals based on discrete *lattices*, *coding* problems, and *multivariate equations* [BBD09, Che16].

## 1.1 Overview of our results

In this paper, we revisit building cryptography via the framework of group actions and aim to provide new candidate and tools that could serve as *quantum-safe* solutions. Our contribution can be summarized in the following three aspects.

First, we propose a family of group actions on *tensors* of order at least three over a finite field as a new candidate for one-way actions. We back up its viability by its relation with other group actions, extensive analysis from heuristic algorithms, provable algorithmic and hardness results, as well as demonstrating its resistance to a standard quantum Fourier sampling technique.

Second, we propose the notion of *pseudorandom group actions* (PRA) that extends the scope of the existing group-action framework. The PRA assumption can be seen as a natural generalization of the Decisional Diffie-Hellman (DDH) assumption. We again instantiate it with the group action on tensors, and we provide evidence (in addition to those for one-wayness) from analyzing various state-of-art attacking strategies.

Finally, based on any PRA, we show realization of several primitives in *minicrypt* such as digital signatures via the Fiat-Shamir transformation and pseudorandom functions. We give complete security proofs against *quantum* adversaries, thanks to recent advances in analyzing quantum *su-*

*perposition* attacks and the quantum random oracle model [Zha12, Unr17, SY17], which is known to be a tricky business. Our constructions based on PRA are more efficient than known schemes based on one-way group actions. As a side contribution, we also describe formal *quantum-security* proofs for several OWA-based schemes including identification and signatures, which are missing in the literature and deserve some care.

In what follows, we elaborate on our proposed group action based on tensors and the new pseudorandom group action assumption. Readers interested in the cryptographic primitives supported by PRA are referred to Section 7.

**The general linear group action on tensors.** The candidate group action we propose is based on *tensors*, a central notion in quantum theory. In this paper, a $k$-tensor $T$ is a multidimensional array with $k$ indices $i_1, i_2, \ldots, i_k$ over a field $\mathbb{F}$, where $i_j \in \{1, 2, \ldots, d_j\}$ for $j = 1, 2, \ldots, k$. For a tuple of indices $(i_1, i_2, \ldots, i_k)$, the corresponding component of $T$ denoted as $T_{i_1,i_2,\ldots,i_k}$ is an element of $\mathbb{F}$. The number $k$ is called the order of the tensor. A matrix over field $\mathbb{F}$ can be regarded as a tensor of order two.

We consider a natural group action on $k$-tensors that represents a local change of basis. Let $G = \prod_{j=1}^{k} \mathrm{GL}(d_j, \mathbb{F})$ be the direct product of general linear groups. For $M = \left(M^{(j)}\right)_{j=1}^{k} \in G$, and a $k$-tensor $T$, the action of $M$ on $T$ is given by

$$\alpha : (M, T) \mapsto \widehat{T}, \text{ where } \widehat{T}_{i_1,i_2,\ldots,i_k} = \sum_{l_1,l_2,\ldots,l_k} \left( \prod_{j=1}^{k} M^{(j)}_{i_j,l_j} \right) T_{l_1,l_2,\ldots,l_k}.$$

We shall refer to the above group action as the *general linear group action on tensors* (GLAT) of dimensions $(d_1, \ldots, d_k)$ over $\mathbb{F}$, or simply GLAT when there is no risk of confusion. We will consider group actions on tensors of order at least three, as the problem is usually easy for matrices. In fact, in most of the cases, we focus on 3-tensors which is most studied and believed to be hard.

**General linear actions on tensors as a candidate for one-way group actions.** We propose to use GLAT as an instantiation of one-way group actions. Roughly speaking, a group action is called a *one-way group action* (OWA in short), if for a random $s \in S$, a random $g \in G$, $t = g \cdot s$, and any polynomial-time adversary $\mathcal{A}$ given $s$ and $t$ as input, $\mathcal{A}$ outputs a $g' \in G$ such that $t = g' \cdot s$ only with negligible probability.

Breaking the one-wayness can be identified with solving some isomorphism problem. Specifically, two $k$-tensors $T$ and $\widehat{T}$ are said to be isomorphic if there exists an $M \in G$ such that $\widehat{T} = \alpha(M, T)$. We define the decisional tensor isomorphism problem (DTI) as deciding if two given $k$-tensors are isomorphic; and the search version (TI) is tasked with computing an $M \in G$ such that $\widehat{T} = \alpha(M, T)$ if there is one. Clearly, our assumption that GLAT is a one-way group action is equivalent to assuming that TI is hard for random $M \in G$, random $k$-tensor $S$, and $T := \alpha(M, S)$. We focus on the case when the order $k$ of the tensor equals three and the corresponding tensor isomorphism problem is abbreviated as 3TI. We justify our proposal from multiple routes; see Section 3 for a more formal treatment.

1. The 3-tensor isomorphism problem can be regarded as "the most difficult" one among problems about testing isomorphism between objects, such as polynomials, graphs, linear codes, and groups, thanks to the recent work of Futorny, Grochow, and Sergeichuk [FGS19]. More specifically, it was proven in [FGS19] that several isomorphism problems, including graph isomorphism, quadratic polynomials with 2 secrets from multivarite cryptography [Pat96],

$p$-group isomorphism from computational group theory [O'B94, LQ17], and linear code permutation equivalence from coding theory [PR97, Sen00], all reduce to 3TI; cf. Observation 2. Note that testing isomorphism of quadratic polynomials with two secrets has been studied in multivariate cryptography for more than two decades [Pat96]. Isomorphism testing of $p$-groups has been studied in computational group theory and theoretical computer science at least since the 1980's (cf. [O'B94, LQ17]). Current status of these two problems then could serve as evidence for the difficulty of 3TI.

2. Known techniques that are effective on GI, including the combinatorial techniques [WL68] and the group-theoretic techniques [Bab79, Luk82], are difficult to translate to 3TI. Indeed, it is not even clear how to adapt a basic combinatorial technique for GI, namely individualizing a vertex [BES80], to the 3TI setting. It is also much harder to work with matrix groups over finite fields than to work with permutation groups. Also, techniques in computer algebra, including those that lead to the recent solution of isomorphism of quadratic polynomials with one secret [IQ19], seem not applicable to 3TI.

3. Finally, there is negative evidence that quantum algorithmic techniques involving the most successful quantum Fourier sampling may not be able to solve GI and code equivalence [HMR$^+$10, DMR15]. It is expected that the same argument holds with respect to 3TI as well. Loosely speaking, this is because the group underlying 3TI is a direct product of general linear groups, which also has irreducible representations of high dimensions.

**A new assumption: pseudorandom group actions.** Inspired by the Decisional Diffie-Hellman assumption, which enables versatile cryptographic constructions, we propose the notion of *pseudorandom group actions*, or PRA in short.

Roughly speaking, we call a group action $\alpha : G \times S \to S$ *pseudorandom*, if any quantum polynomial-time algorithm $\mathcal{A}$ cannot distinguish the following two distributions except with negligible probability: $(s, t)$ where $s, t \in_R S$, and the other distribution $(s, \alpha(g, s))$, where $s \in_R S$ and $g \in_R G$. A precise definition can be found in Section 4.

Note that if a group action is transitive, then the pseudorandom distribution trivially coincides with the random distribution. Unless otherwise stated, we will consider *intransitive* group actions when working with pseudorandom group actions. In fact, we can assume that $(s, t)$ from the random distribution are in different orbits with high probability, while $(s, t)$ from the pseudorandom distribution are always in the same orbit.

Also note that PRA is a stronger assumption than OWA. To break PRA, it is enough to solve the isomorphism testing problem *on average* in a relaxed sense, i.e., on $1/\operatorname{poly}(n)$ fraction of the input instances instead of all but $1/\operatorname{poly}(n)$ fraction, where $n$ is the input size.

The Decisional Diffie-Hellman (DDH) assumption [DH76, Bon98] can be seen as the PRA initiated with a certain group action; see Observation 4. However, DDH is broken on a quantum computer. We resort again to GLAT as a quantum-safe candidate of PRA. We investigate the hardness of breaking PRA from various perspectives and provide further justification for using the general linear action on 3-tensors as a candidate for PRA.

1. Easy instances on 3-tensors seem scarce, and average-case algorithms do not speed up dramatically. Indeed, the best known average-case algorithm, while improves over worst-case somewhat due to the birthday paradox, still inherently enumerate all vectors in $\mathbb{F}_q^n$ and hence take exponential time [BFV13, LQ17].

2. For 3-tensors, there have not been non-trivial and easy-to-compute isomorphism invariants, i.e., those properties that are preserved under the action. For example, a natural isomorphism invariant, the tensor rank, is well-known to be NP-hard [Hås90]. Later work suggests that "most tensor problems are NP-hard" [HL13].

3. We propose and analyze several attack strategies from group theory and geometry. While effective on some non-trivial actions, these attacks do not work for the general linear action on 3-tensors. For instance, we notice that breaking our PRA from GLAT reduces to the orbit closure intersection problem, which has received considerable attention in optimization, and geometric complexity theory. Despite recent advances [Mul17, BGdO$^+$18, BFG$^+$18, AGL$^+$18, DM18, IQS17], any improvement towards a more effective attack would be a breakthrough.

Recently, De Feo and Galbraith proposed an assumption in the setting of supersingular isogeny-based cryptography, which can be viewed as another instantiation of PRA [FG19, Problem 4]. This gives more reason to further explore PRA as a basic building block in cryptography.

### 1.2 Discussions

In this paper, we further develop and extend the scope of group action based cryptography by introducing the general linear group actions on tensors, GLAT, to the family of instantiations, by formulating the pseudorandom assumption generalizing the well-known DDH assumption, and by proving the quantum-security of various cryptographic primitives such as signatures and pseudo-random functions in this framework.

There are two key features of GLAT that are worth mentioning explicitly. First, the general linear action is *non-commutative* simply because the general linear group is non-abelian. This is, on the one hand, an attractive property that enabled us to argue the quantum hardness and the infeasibility of quantum Fourier sampling type of attacks. On the other hand, however, this also makes it challenging to extend many attractive properties of discrete-logarithm and decisional Diffie-Hellman to the more general framework of group action cryptography. For example, while it is known that the worst-case DDH assumption reduces to the average-case DDH assumption [NR04], the proof relies critically on commutativity. Second, the general linear action is *linear* and the space of tensors form a linear space. Linearity seems to be responsible for the supergroup attacks on the PRA($d$) assumption discussed in Subsection 5.1.1. It also introduces the difficulty for building more efficient PRF constructions analogous to the DDH-based ones proposed in [NR04].

Our work leaves a host of basic problems about group action based cryptography as future work. First, we have been focusing on the general linear group actions on tensors and have not discussed too much about the other possible group actions on the tensor space. A mixture of different types of group actions on different indices of the tensor may have the advantage of obtaining a more efficient construction or other appealing structural properties. It will be interesting to understand better how the hardness of the group actions on tensors relate to each other and what are the good choices of group actions for practicability considerations. Second, it is appealing to recover the average-case to worst-case reduction, at least to some extent, for the general group actions framework. Finally, it is an important open problem to build quantum-secure public-key encryption schemes based on hard problems about GLAT or its close variations.

## 2 The group action framework

In this section, we formally describe the framework for group action based cryptography to be used in this paper. While such general frameworks were already proposed by Brassard and Yung [BY90]

and Couveignes [Cou06], there are delicate differences in several places, so we will have to still go through the details. This section should be considered as largely expository.

### 2.1 Group actions and notations

Let us first formally define group actions. Let $G$ be a group, $S$ be a set, and id the identity element of $G$. A *(left) group action* of $G$ on $S$ is a function $\alpha : G \times S \to S$ satisfying the following: (1) $\forall s \in S$, $\alpha(\mathrm{id}, s) = s$; (2) $\forall g, h \in G$, $s \in S$, $\alpha(gh, s) = \alpha(g, \alpha(h, s))$. The group operation is denoted by $\circ$, e.g. for $g, h \in G$, we can write their product as $g \circ h$. We shall use $\cdot$ to denote the left action, e.g. $g \cdot s = \alpha(g, s)$. We may also consider the right group action $\beta : S \times G \to S$, and use the exponent notation for right actions, e.g. $s^g = \beta(s, g)$.

Later, we will use a special symbol $\perp \notin G \cup S$ to indicate that a bit string does not correspond to an encoding of an element in $G$ or $S$. We extend the operators $\circ$ and $\cdot$ to $\circ : G \cup \{\perp\} \times G \cup \{\perp\} \to G \cup \{\perp\}$ and $\cdot : G \cup \{\perp\} \times S \cup \{\perp\} \to S \cup \{\perp\}$, by letting $g \circ h = \perp$ whenever $g = \perp$ or $h = \perp$, and $g \cdot s = \perp$ whenever $g = \perp$ or $s = \perp$.

Let $\alpha : G \times S \to S$ be a group action. For $s \in S$, the *orbit* of $s$ is $O_s = \{t \in S : \exists g \in G, g \cdot s = t\}$. The action $\alpha$ partitions $S$ into a disjoint union of orbits. If there is only one orbit, then $\alpha$ is called transitive. Restricting $\alpha$ to any orbit $O$ gives a transitive action. In this case, take any $s \in O$, and let $\mathrm{Stab}(s, G) = \{g \in G : g \cdot s = s\}$ be the stabilizer group of $s$ in $G$. For any $t \in O$, those group elements sending $s$ to $t$ form a coset of $\mathrm{Stab}(s, G)$. We then obtain the following easy observation.

**Observation 1.** Let $\alpha : G \times S \to S$, $s$, and $O$ be as above. The following two distributions are the same: the uniform distribution of $t \in O$, and the distribution of $g \cdot s$ where $g$ is sampled from a uniform distribution over $G$.

### 2.2 The computational model

For computational purposes, we need to model the algorithmic representations of groups and sets, as well as basic operations like group multiplication, group inverse, and group actions. We review the group action framework as proposed in Brassard and Yung [BY90]. A variant of this framework, with a focus on restricting to abelian (commutative) groups, was studied by Couveignes [Cou06]. However, it seems to us that some subtleties are present, so we will propose another version, and compare it with those by Brassard and Yung, and Couveignes, later.

- Let $n$ be a parameter which controls the instance size. Therefore, polynomial time or length in the following are with respect to $n$.
- (Representing group and set elements.) Let $G$ be a group, and $S$ be a set. Let $\alpha : G \times S \to S$ be a group action. Group elements and set elements are represented by bit strings $\{0,1\}^*$. There are polynomials $p(n)$ and $q(n)$, such that we only work with group elements representable by $\{0,1\}^{p(n)}$ and set elements representable by $\{0,1\}^{q(n)}$. There are functions $F_G$ and $F_S$ from $\{0,1\}^*$ to $G \cup \{\perp\}$ and $S \cup \{\perp\}$, respectively. Here, $\perp$ is a special symbol, designating that the bit string does not represent a group or set element. $F_G$ and $F_S$ should be thought of as assigning bit strings to group elements.
- (Unique encoding of group and set elements.) For any $g \in G$, there exists a unique $b \in \{0,1\}^*$ such that $F_G(b) = g$. In particular, there exists a unique bit string, also denoted by id, such that $F_G(\mathrm{id}) = \mathrm{id}$. Similarly, for any $s \in S$, there exists a unique $b \in \{0,1\}^*$ such that $F_S(b) = s$.

- (Group operations.) There are polynomial-time computable functions PROD : $\{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ and INV : $\{0,1\}^* \to \{0,1\}^*$, such that for $b, c \in \{0,1\}^*$, $F_G(\text{PROD}(b,c)) = F_G(b) \circ F_G(c)$, and $F_G(\text{INV}(b)) \circ F_G(b) = \text{id}$.
- (Group action.) There is a polynomial-time function $a : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$, such that for $b \in \{0,1\}^*$ and $c \in \{0,1\}^*$, satisfies $F_S(a(b,c)) = \alpha(F_G(b), F_S(c))$.
- (Recognizing group and set elements.) There are polynomial-time computable functions $C_G$ and $C_S$, such that $C_G(b) = 1$ iff $F_G(b) \neq \perp$, and $C_S(b) = 1$ iff $F_S(b) \neq \perp$.
- (Random sampling of group and set elements.) There are polynomial-time computable functions $R_G$ and $R_S$, such that $R_G$ uniformly samples a group element $g \in G$, represented by the unique $b \in \{0,1\}^{p(n)}$ with $F_G(b) = g$, and $R_S$ uniformly samples a set element $s \in S$, represented by some $b \in \{0,1\}^{q(n)}$ with $F_S(b) = s$.

**Remark 1.** Some remarks are due for the above model.

1. The differences with Brassard and Yung are: (1) allowing infinite groups and sets; (2) adding random sampling of set elements. Note that in the case of infinite groups and sets, the parameters $p(n)$ and $q(n)$ are used to control the bit lengths for the descriptions of legitimate group and set elements. This allows us to incorporate e.g. the lattice isomorphism problem [HR14] into this framework. In the rest of this article, however, we will mostly work with finite groups and sets, unless otherwise stated.
2. The main reason to consider infinite groups is the uses of lattice isomorphism and equivalence of integral bilinear forms in the cryptographic setting.
3. The key difference with Couveignes lies in Couveignes's focus on transitive abelian group actions with trivial stabilizers.
4. It is possible to adapt the above framework to use the black-box group model by Babai and Szemerédi [BS84], whose motivation was to deal with non-unique encodings of group elements (like quotient groups). For our purposes, it is more convenient and practical to assume that the group elements have unique encodings.
5. Babai [Bab91] gives an efficient Monte Carlo algorithm for sampling a group element of a finite group in a very general setting which is applicable to most of our instantiations with finite groups.

### 2.3 The isomorphism problem and the one-way assumption

Now that we have defined group actions and a computational model, let us examine the isomorphism problems associated with group actions.

**Definition 2** (The isomorphism problem)**.** Let $\alpha : G \times S \to S$ be a group action. The isomorphism problem for $\alpha$ is to decide, given $s, t \in S$, whether $s$ and $t$ lie in the same orbit under $\alpha$. If they are, the search version of the isomorphism problem further asks to compute some $g \in G$, such that $\alpha(g, s) = t$.

If we assume that there is a distribution on $S$ and we require the algorithm to succeed for $(s, t)$ where $s$ is sampled from this distribution and $t$ is arbitrary, then this is the *average-case* setting

of the isomorphism problem. For example, the first average-case efficient algorithm for the graph isomorphism problem was designed by Babai, Erdős and Selkow in the 1970's [BES80].

The hardness of the isomorphism problem provides us with the basic intuition for its use in cryptography. But for cryptographic uses, the *promised* search version of the isomorphism problem is more relevant, as already observed by Brassard and Yung [BY90]. That is, suppose we are given $s, t \in S$ with the promise that they are in the same orbit, the problem asks to compute $g \in G$ such that $g \cdot s = t$. Making this more precise and suitable for cryptographic purposes, we formulate the following problem.

**Definition 3** (The group-action inversion (GA-Inv) problem)**.** Let $\mathcal{G}$ be a group action family, such that for a security parameter $\lambda$, $\mathcal{G}(1^\lambda)$ consists of descriptions of a group $G$, a set $S$ with $\log(|G|) = \text{poly}(\lambda)$, $\log(|S|) = \text{poly}(\lambda)$, and an group action $\alpha : G \times S \to S$ that can be computed efficiently, which we denote as a whole as a public parameter `params`. Generate random $s \leftarrow S$ and $g \leftarrow G$, and compute $t := \alpha(g, s)$. The group-action inversion (GA-Inv) problem is to find $g$ given $(s, t)$.

**Definition 4** (Group-action inversion game)**.** The group-action inversion game is the following game between a challenger and an arbitrary adversary $\mathcal{A}$:

1. The challenger and adversary $\mathcal{A}$ agree on the public parameter `params` by choosing it to be $\mathcal{G}(1^\lambda)$ for some security parameter $\lambda$.
2. Challenger samples $s \leftarrow S$ and $g \leftarrow G$ using $R_S$ and $R_G$, computes $t = g \cdot s$, and gives $(s, t)$ to $\mathcal{A}$.
3. The adversary $\mathcal{A}$ produces some $g'$ and sends it to the challenger.
4. We define the output of the game $\mathsf{GA\text{-}Inv}_{\mathcal{A},\mathcal{G}}(1^\lambda) = 1$ if $g' \cdot s = t$, and say $\mathcal{A}$ wins the game if $\mathsf{GA\text{-}Inv}_{\mathcal{A},\mathcal{G}}(1^\lambda) = 1$.

**Definition 5.** We say that the group-action inversion (GA-Inv) problem is hard relative to $\mathcal{G}$, if for any polynomial time quantum algorithm $\mathcal{A}$,

$$\Pr\left[\mathsf{GA\text{-}Inv}_{\mathcal{A},\mathcal{G}}(1^\lambda)\right] \leq \text{negl}(\lambda)\,.$$

We propose our first cryptographic assumption in the following. It generalizes the one in [BY90].

**Assumption 1** (One-way group action (OWA) assumption)**.** There exists a family $\mathcal{G}$ relative to which the GA-Inv problem is hard.

We informally call the group action family $\mathcal{G}$ in Assumption 1 a one-way group action. Its name comes from the fact that, as already suggested in [BY90], this assumption immediately implies that we can treat $\Gamma_s : G \to S$ given by $\Gamma_s(g) = \alpha(g, s)$ as a one-way function for a random $s$. In fact, OWA assumption is equivalent to the assertion that the function $\Gamma : G \times S \to S \times S$ given by $\Gamma(g, s) = (g \cdot s, s)$ is one-way in the standard sense.

Note that the OWA assumption comes with the promise that $s$ and $t$ are in the same orbit. The question is to compute a group element that sends $s$ to $t$. Comparing with Definition 2, we see that the OWA assumption is stronger than the assumption that the search version of the isomorphism problem is hard for a group action, while incomparable with the decision version. Still, most algorithms for the isomorphism problem we are aware of do solve the search version.

**Remark 6.** Note that Assumption 1 has a slight difference with that of Brassard and Yung as follows. In [BY90], Brassard and Yung asks for the *existence* of some $s \in S$ as in Definition 3, such that for a random $g \in G$, it is not feasible to compute $g'$ that sends $s$ to $\alpha(g, s)$. Here, we relax this condition, namely a *random* $s \in S$ satisfies this already. One motivation for Brassard and Yung to fix $s$ was to take into account of graph isomorphism, for which Brassard and Crepéau defined the notion of "hard graphs" which could serve as this starting point [BC86]. However, by Babai's algorithm [Bab16] we know that hard graphs could not exist. Here we use a stronger notion by allowing a random $s$, which we believe is a reasonable requirement for some concrete group actions discussed in Section 3.

A useful fact for the GA-Inv problem is that it is *self-reducible* to random instances within the orbit of the input pair. For any given $s$, let $O_s$ be the orbit of $s$ under the group action $\alpha$. If there is an efficient algorithm $\mathcal{A}$ that computes $g$ from $(t, t')$ where $t' = \alpha(g, t)$ for at least $1/\operatorname{poly}(\lambda)$ fraction of the pairs $(t, t') \in O_s \times O_s$, then the GA-Inv problem can be computed for any $(t, t') \in O_s \times O_s$ with probability $1 - e^{-\operatorname{poly}(\lambda)}$. On input $(t, t')$, the algorithm samples random group elements $h, h'$ and calls $\mathcal{A}$ with $(\alpha(h, t), \alpha(h', t'))$. If $\mathcal{A}$ successfully returns $g$, the algorithm outputs $h^{-1}gh'$ and otherwise repeats the procedure for polynomial number of times.

The one-way assumption leads to several basic cryptographic applications as described in the literature. First, it gives a identification scheme by adapting the zero-knowledge proof system for graph isomorphism [GMW91]. Then via the celebrated Fiat-Shamir transformation [FS86], one also obtains a signature scheme. Proving quantum security of these protocols, however, would need more care. For completeness, we give detailed proofs in Section 6.

# 3 General linear actions on tensors: the one-way group action assumption

In this section, we propose the general linear actions on tensors, i.e., the tensor isomorphism problem, as our choice of candidate for the OWA assumption. We first reflect on what would be needed for a group action to be a good candidate.

## 3.1 Requirements for a group action to be one-way

Naturally, the hardness of the GA-Inv problem for a specific group action needs to be examined in the context of the following four types of algorithms.

- Practical algorithms: implemented algorithms with practical performance evaluations but no theoretical guarantees;
- Average-case algorithms: for some natural distribution over the input instances, there is an algorithm that are efficient for most input instances from this distribution with provable guarantees;
- Worst-case algorithms: efficient algorithms with provable guarantees for all input instances;
- Quantum algorithms: average-case or worst-case efficient algorithms in the quantum setting.

Here, efficient means sub-exponential, and most means $1 - 1/\operatorname{poly}(n)$ fraction. It is important to keep in mind all possible attacks by these four types of algorithms. Past experience suggests that one problem may look difficult from one viewpoint, but turns out to be easy from another.

The graph isomorphism problem has long been thought to be a difficult problem from the worst-case viewpoint. Indeed, a quasipolynomial-time algorithm was only known very recently, thanks to Babai's breakthrough [Bab16]. However, it has long been known to be effectively solvable from the practical viewpoint [McK80, MP14]. This shows the importance of practical algorithms when justifying a cryptographic assumption.

Patarin proposed to use polynomial map isomorphism problems in his instantiation of the identification and signature schemes [Pat96]. He also proposed the one-sided version of such problems, which has been studied intensively [PGC98, GMS03, Per05, FP06, Kay11, BFFP11, MPG13, BFV13, PFM14, BFP15], mostly from the viewpoint of practical cryptanalysis. However, the problem of testing isomorphism of quadratic polynomials with one secret was recently shown to be solvable in randomized polynomial time [IQ19], using ideas including efficient algorithms for computing the algebra structure, and the $*$-algebra structure underlying such problems. Hence, the investigation of theoretical algorithms is also valuable.

Considering of quantum attacks is necessary for security in the quantum era. Shor's algorithm, for example, invalidates the hardness assumption of the discrete logarithm problems.

Guided by the difficulty met by the hidden subgroup approach on tackling graph isomorphism [HMR$^+$10], Moore, Russell, and Vazirani proposed the code equivalence problem as a candidate for the one-way assumption [MRV07]. However, this problem turns out to admit an effective practical algorithm by Sendrier [Sen00].

### 3.1.1 One-way group action assumption and the hidden subgroup approach

From the post-quantum perspective, a general remark can be made on the OWA assumption and the hidden subgroup approach in quantum algorithm design.

Recall that the hidden subgroup approach is a natural generalization of Shor's quantum algorithms for discrete logarithm and factoring [Sho94], and can accommodate both lattice problems [Reg04] and isomorphism testing problems [HMR$^+$10]. The survey paper of Childs and van Dam [CvD10] contains a nice introduction to this approach.

A well-known approach to formulate GA-Inv as an HSP problem is the following [CvD10, Sec. VII.A]. Let $\alpha : G \times S \to S$ be a group action. Given $s, t \in S$ with the promise that $t = g \cdot s$ for some $g \in G$, we want to compute $g$. To cast this problem as an HSP instance, we first formulate it as an automorphism type problem. Let $\tilde{G} = G \wr \mathrm{S}_2$, where $\mathrm{S}_2$ is the symmetric group on two elements, and $\wr$ denotes the wreath product. The action $\alpha$ induces an action $\beta$ of $\tilde{G}$ on $S \times S$ as follows. Given $(g, h, i) \in \tilde{G} = G \wr \mathrm{S}_2$ where $g, h \in G, i \in \mathrm{S}_2$, if $i$ is the identity, it sends $(s, t) \in S \times S$ to $(g \cdot s, h \cdot t)$; otherwise, it sends $(s, t)$ to $(h \cdot t, g \cdot s)$. Given $(s, t) \in S \times S$, we define a function $f_{(s,t)} : \tilde{G} \to S \times S$, such that $f_{(s,t)}$ sends $(g, h, i)$ to $(g, h, i) \cdot (s, t)$, defined as above. It can be verified that $f_{(s,t)}$ hides the coset of the stabilizer group of $(s, t)$ in $\tilde{G}$. Since $s$ and $t$ lie in the same orbit, any generating set of the stabilizer group of $(s, t)$ contains an element of the form $(g, h, i)$, where $i$ is not the identity element in $\mathrm{S}_2$, $g \cdot s = t$, and $h \cdot t = s$. In particular, $g$ is the element required to solve the GA-Inv problem. In the above reduction to the HSP problem, the ambient group is $G \wr \mathrm{S}_2$ instead of the original $G$. In some cases like the graph isomorphism problem, because of the polynomial-time reduction from isomorphism testing to automorphism problem, we can retain the ambient group to be $G$. However, such a reduction is not known for GLAT.

There has been notable progress on the HSP problems for various ambient groups, but the dihedral groups and the symmetric groups have withstood the attacks so far. Indeed, one source of confidence on using lattice problems in post-quantum cryptography lies in the lack of progress in tackling the hidden subgroup problem for dihedral groups [Reg04]. There is formal negative evidence for the applicability of this approach for certain group actions where the groups have

high-dimensional representations, like $\mathrm{S}_n$ and $\mathrm{GL}(n, q)$ in the case of the graph isomorphism problem [HMR+10] and the permutation code equivalence problem [DMR15]. The general lesson is that current quantum algorithmic technologies seem incapable of handling groups which have irreducible representations of high dimensions.

As mentioned, the OWA assumption has been discussed in post-quantum cryptography with the instantiation of the permutation code equivalence problem [MRV07, DMR11a, DMR11b, SS13, DMR15]. Though this problem is not satisfying enough due to the existence of effective practical algorithms [Sen00], the following quoted from [MRV07] would be applicable to our choice of candidate to the discussed below.

> *The design of efficient cryptographic primitives resistant to quantum attack is a pressing practical problem whose solution can have an enormous impact on the practice of cryptography long before a quantum computer is physically realized. A program to create such primitives must necessarily rely on insights into the limits of quantum algorithms, and this paper explores consequences of the strongest such insights we have about the limits of quantum algorithms.*

## 3.2 The tensor isomorphism problem and others

We now formally define the tensor isomorphism problem and other isomorphism testing problems. For this we need some notation and preparations.

### 3.2.1 Notation and preliminaries

We usually use $\mathbb{F}$ to denote a field. The finite field with $q$ elements and the real number field are denoted by $\mathbb{F}_q$ and $\mathbb{R}$, respectively. The linear space of $m$ by $n$ matrices over $\mathbb{F}$ is denoted by $\mathrm{M}(m, n, \mathbb{F})$, and $\mathrm{M}(n, \mathbb{F}) := \mathrm{M}(n, n, \mathbb{F})$. The identity matrix in $\mathrm{M}(n, \mathbb{F})$ is denoted by $I_n$. For $A \in \mathrm{M}(m, n, \mathbb{F})$, $A^t$ denotes the transpose of $A$. The group of $n$ by $n$ invertible matrices over $\mathbb{F}$ is denoted by $\mathrm{GL}(n, \mathbb{F})$. We will also meet the notation $\mathrm{GL}(n, \mathbb{Z})$, the group of $n$ by $n$ integral matrices with determinant $\pm 1$. We use a slightly non-standard notation $\mathrm{GL}(m, n, \mathbb{F})$ to denote the set of rank $\min(m, n)$ matrices in $\mathrm{M}(m, n, \mathbb{F})$. We use $\langle \cdot \rangle$ to denote the linear span; for example, given $A_1, \ldots, A_k \in \mathrm{M}(m, n, \mathbb{F})$, $\langle A_1, \ldots, A_k \rangle$ is a subspace of $\mathrm{M}(m, n, \mathbb{F})$.

We will meet some subgroups of $\mathrm{GL}(n, \mathbb{F})$ as follows. The symmetric group $\mathrm{S}_n$ on $n$ objects is embedded into $\mathrm{GL}(n, \mathbb{F})$ as permutation matrices. The orthogonal group $\mathrm{O}(n, \mathbb{F})$ consists of those invertible matrices $A$ such that $A^t A = I_n$. The special linear group $\mathrm{SL}(n, \mathbb{F})$ consists of those invertible matrices $A$ such that $\det(A) = 1$. Finally, when $n = \ell^2$, there are subgroups of $\mathrm{GL}(\ell^2, \mathbb{F})$ isomorphic to $\mathrm{GL}(\ell, \mathbb{F}) \times \mathrm{GL}(\ell, \mathbb{F})$. This can be seen as follows. First we fix an isomorphism of linear spaces $\phi : \mathbb{F}^{\ell^2} \to \mathrm{M}(\ell, \mathbb{F})$[1]. Then $\mathrm{M}(\ell, \mathbb{F})$ admits an action by $\mathrm{GL}(\ell, \mathbb{F}) \times \mathrm{GL}(\ell, \mathbb{F})$ by left and right multiplications, e.g. $(A, D) \in \mathrm{GL}(\ell, \mathbb{F}) \times \mathrm{GL}(\ell, \mathbb{F})$ sends $C \in \mathrm{M}(\ell, \mathbb{F})$ to $ACD^t$. Now use $\phi^{-1}$ and we get one subgroup of $\mathrm{GL}(\ell^2, \mathbb{F})$ isomorphic to $\mathrm{GL}(\ell, \mathbb{F}) \times \mathrm{GL}(\ell, \mathbb{F})$.

### 3.2.2 Definitions of several group actions

We first recall the concept of tensors and the group actions on the space of $k$-tensors as introduced in Section 1.

**Definition 7** (Tensor)**.** A $k$-tensor $T$ of local dimensions $d_1, d_2, \ldots, d_k$ over $\mathbb{F}$, written as

$$T = (T_{i_1, i_2, \ldots, i_k}),$$

[1]For example, we can let the first $\ell$ components be the first row, the second $\ell$ components be the second row, etc..

is a multidimensional array with $k$ indices and its components $T_{i_1,i_2,\ldots,i_k}$ chosen from $\mathbb{F}$ for all $i_j \in \{1, 2, \ldots, d_j\}$. The set of $k$-tensors of local dimensions $d_1, d_2, \ldots, d_k$ over $\mathbb{F}$ is denoted as

$$\mathrm{T}(d_1, d_2, \ldots, d_k, \mathbb{F}).$$

The integer $k$ is called the order of tensor $T$.

**Group Action 1** (The general linear group action on tensors)**.** Let $\mathbb{F}$ be a field, $k$, $d_1, d_2, \ldots, d_k$ be integers.

- Group $G$: $\prod_{j=1}^{k} \mathrm{GL}(d_j, \mathbb{F})$.
- Set $S$: $\mathrm{T}(d_1, d_2, \ldots, d_k, \mathbb{F})$.
- Action $\alpha$: for a $k$-tensor $T \in S$, a member $M = (M^{(1)}, M^{(2)}, \ldots, M^{(k)})$ of the group $G$,

$$\alpha(M, T) = \left(\bigotimes_{j=1}^{k} M^{(j)}\right) T = \sum_{l_1,l_2,\ldots,l_k} \left(\prod_{j=1}^{k} M^{(j)}_{i_j,l_j}\right) T_{l_1,l_2,\ldots,l_k}.$$

We refer to the general linear group action on tensors in Action 1 as GLAT. In the following, let us formally define several problems which have been referred to frequently in the above discussions.

As already observed by Brassard and Yung [BY90], the discrete logarithm problem can be formulated using the language of group actions. More specifically, we have:

**Group Action 2** (Discrete Logarithm in Cyclic Groups of Prime Orders)**.** Let $p$ be a prime, $\mathbb{Z}_p$ the integer.

- Group $G$: $\mathbb{Z}_p^*$, the multiplicative group of units in $\mathbb{Z}_p$.
- Set $S$: $C_p \setminus \{\mathrm{id}\}$, where $C_p$ is a cyclic group of order $p$ and id is the identity element.
- Action $\alpha$: for $a \in \mathbb{Z}_p^*$, and $s \in S$, $\alpha(a, s) = s^a$.

Note that in the above, we refrained from giving a specific realization of the cyclic group $C_p$ for the sake of clarify; the reader may refer to Boneh's excellent survey [Bon98] for concrete proposals that can support the security of the Decisional Diffie-Hellman assumption.

The linear code permutation equivalence (LCPE) problem asks to decide whether two linear codes (i.e. linear subspaces) are the same up to a permutation of the coordinates. It has been studied in the coding theory community since the 1990's [PR97, Sen00].

**Group Action 3** (Group action for Linear Code Permutation Equivalence problem (LCPE))**.** Let $m, d$ be integers, $m \leq d$, and let $\mathbb{F}$ be a field.

- Group $G$: $\mathrm{GL}(m, \mathbb{F}) \times \mathrm{S}_d$.
- Set $S$: $\mathrm{GL}(m, d, \mathbb{F})$.
- Action $\alpha$: for $A \in S$, $M = (N, P) \in G$, $\alpha(M, A) = NAP^t$.

The connection with coding theory is that $A$ can be viewed as the generating matrix of a linear code (a subspace of $\mathbb{F}_q^n$), and $N$ is the change of basis matrix taking care of different choices of bases. Then, $P$, as a permutation matrix, does not change the weight of a codeword— that is a vector in $\mathbb{F}^n$. (There are other operations that preserve weights [SS13], but we restrict to consider this setting

for simplicity.) The GA-Inv problem for this group action is called the linear code permutation equivalence (LCPE) problem, which has been studied in the coding theory community since the 1980's [Leo82], and we can dodge the only successful attack [Sen00] by restricting to self-dual codes.

The following group action induces a problem called the polynomial isomorphism problems proposed by Patarin [Pat96], and has been studied in the multivariate cryptography community since then.

**Group Action 4** (Group action for the Isomorphism of Quadratic Polynomials with two Secrets problem (IQP2S))**.** Let $m, d$ be integers and $\mathbb{F}$ a finite field.

- Group $G$: $\mathrm{GL}(d, \mathbb{F}) \times \mathrm{GL}(m, \mathbb{F})$.
- Set $S$: The set of tuples of homogeneous polynomials $(f_1, f_2, \ldots, f_m)$ for $f_i \in \mathbb{F}[x_1, x_2, \ldots, x_d]$ the polynomial ring of $d$ variables over $\mathbb{F}$.
- Action $\alpha$: for $f = (f_1, f_2, \ldots, f_m) \in S$, $M = (C, D) \in G$, $C' = C^{-1}$, define $\alpha(M, f) = (g_1, g_2, \ldots, g_m)$ by $g_i(x_1, x_2, \ldots, x_d) = \sum_{j=1}^{m} D_{i,j} f_i(x'_1, \ldots, x'_d)$, where $x'_i = \sum_{j=1}^{d} C'_{i,j} x_j$.

The GA-Inv problem for this group action is essentially the isomorphism of quadratic polynomials with two secrets (IQP2S) assumption. The algebraic interpretation here is that the tuple of polynomials $(f_1, \ldots, f_n)$ is viewed as a polynomial map from $\mathbb{F}^n$ to $\mathbb{F}^m$, by sending $(a_1, \ldots, a_n)$ to $(f_1(a_1, \ldots, a_n), \ldots, f_m(a_1, \ldots, a_n))$. The changes of bases by $C$ and $D$ then are naturally interpreted as saying that the two polynomial maps are essentially the same.

Finally, the GA-Inv problem for the following group action originates from computational group theory, and is basically equivalent to a bottleneck case of the group isomorphism problem (i.e. $p$-groups of class 2 and exponent $p$) [O'B94, LQ17].

**Group Action 5** (Group action for alternating matrix space isometry (AMSI))**.** Let $d, m$ be integers and $\mathbb{F}$ be a finite field.

- Group $G$: $\mathrm{GL}(m, \mathbb{F})$.
- Set $S$: the set of all linear spans $\mathcal{A}$ of $d$ alternating[2] matrices $A_i$ of size $m \times m$.
- Action $\alpha$: for $\mathcal{A} = \langle A_1, A_2, \ldots, A_d \rangle \in S$, $C \in G$, $\alpha(C, \mathcal{A}) = \langle B_1, B_2, \ldots, B_d \rangle$ where $B_i = CA_iC^t$ for all $i = 1, 2, \ldots, d$.

## 3.3 General linear actions on tensors as one-way action candidates

### 3.3.1 The central position of 3-tensor isomorphism

As mentioned, the four problems, linear code permutation equivalence (LCPE), isomorphism of polynomials with two secrets (IQP2S), and alternating matrix space isometry (AMSI), have been studied in coding theory, multivariate cryptography, and computational group theory, respectively, for decades. Only recently we begin to see connections among these problems which go through the 3TI problem thanks to the work of Futorny, Grochow, and Sergeichuk [FGS19]. We spell out this explicitly.

**Observation 2** ([FGS19, Gro12])**.** IQP2S, AMSI, GI, and LCPE reduce to 3TI.

[2] An $m \times m$ matrix $A$ is alternating if for any $v \in \mathbb{F}^n$, $v^t A v = 0$.

*Proof.* Note that the set underlying Group Action 5 consists of $d$-tuples of $m \times m$ alternating matrices. We can write such a tuple $(A_1, \ldots, A_d)$ as a 3-tensor $A$ of dimension $m \times m \times d$, such that $A_{i,j,k} = (A_k)_{i,j}$. Then AMSI asks to test whether two such 3-tensors are in the same orbit under the action of $(M, N) \in \mathrm{GL}(m, \mathbb{F}) \times \mathrm{GL}(d, \mathbb{F})$ by sending a 3-tensor $A$ to the result of applying $(M, M, N)$ to $A$ as in the definition of GLAT.

Such an action belongs to the class of actions on 3-tensors considered in [FGS19] under the name *linked actions*. This work constructs a function $r$ from 3-tensors to 3-tensors, such that $A$ and $B$ are in the same orbit under $\mathrm{GL}(m, \mathbb{F}) \times \mathrm{GL}(d, \mathbb{F})$ if and only if $r(A)$ and $r(B)$ are in the same orbit under $\mathrm{GL}(m, \mathbb{F}) \times \mathrm{GL}(m, \mathbb{F}) \times \mathrm{GL}(d, \mathbb{F})$. This function $r$ can be computed efficiently [FGS19, Remark 1.1].

This explains the reduction of the isomorphism problem for Group Action 5 to the 3-tensor isomorphism problem. For Group Action 4, by using the classical correspondence between homogeneous quadratic polynomials and symmetric matrices, we can cast it in a form similar to Group Action 5, and then apply the above reasoning using again [FGS19].

Finally, to reduce the graph isomorphism problem (GI) and the linear code permutation equivalent problem (LCPE) to the 3-tensor isomorphism problem, we only need to take care of LCPE as GI reduces to LCPE [PR97]. To reduce LCPE to 3TI, we can reduce it to the matrix Lie algebra conjugacy problem by [Gro12], which reduces to 3TI by [FGS19] along the linked action argument, though this time linked in a different way. □

This put 3TI at a central position of these difficult isomorphism testing problems arising from multivariate cryptography, computational group theory, and coding theory. In particular, from the worst-case analysis viewpoint, 3TI is the hardest problem among all these. This also allows us to draw experiences from previous research in various research communities to understand 3TI.

### 3.3.2 Current status of the tensor isomorphism problem and its one-way action assumption

We now explain the current status of the tensor isomorphism problem to support it as a strong candidate for the OWA assumption. Because of the connections with isomorphism of polynomials with two secrets (IQP2S) and alternating matrix space isometry (AMSI), we shall also draw results and experiences from the multivariate cryptography and the computational group theory communities.

For convenience, we shall restrict to finite fields $\mathbb{F}_q$, though other fields are also interesting. That is, we consider the action of $\mathrm{GL}(\ell, \mathbb{F}_q) \times \mathrm{GL}(n, \mathbb{F}_q) \times \mathrm{GL}(m, \mathbb{F}_q)$ on $T \in \mathrm{T}(\ell, n, m, \mathbb{F}_q)$. Without loss of generality, we assume $\ell \geq n \geq m$. The reader may well think of the case when $\ell = n = m$, which seems to be the most difficult case in general. Correspondingly, we will assume that the instances for IQP2S are $m$-tuples of homogeneous quadratic polynomials in $n$ variables over $\mathbb{F}_q$, and the instances for AMSI are $m$-tuples of alternating matrices of size $n \times n$ over $\mathbb{F}_q$.

To start, we note that 3TI over finite fields belongs to $\mathrm{NP} \cap \mathrm{coAM}$, following the same coAM-protocol for graph isomorphism.

For the worst-case time complexity, it can be solved in time $q^{m^2} \cdot \mathrm{poly}(\ell, m, n, \log q)$, by enumerating $\mathrm{GL}(m, q)$, and then solving an instance of the matrix tuple equivalence problem, which asks to decide whether two matrix tuples are the same under the left-right multiplications of invertible matrices. This problem can be solved in deterministic polynomial time by reducing [IQ19] to the module isomorphism problem, which in turn admits a deterministic polynomial-time solution [CIK97, BL08, IKS10]. It is possible to reduce the complexity to $q^{cm^2} \cdot \mathrm{poly}(\ell, m, n, \log q)$ for some constant $0 < c < 1$, by using some dynamic programming technique as in [LQ17]. But in gen-

eral, the worst-case complexity could not go beyond this at present, which matches the experiences of IQP2S and AMSI as well; see [IQ19].

For the average-case time complexity, it can be solved in time $q^{O(m)} \cdot \mathrm{poly}(\ell, n)$, by adapting the average-case algorithm for AMSI in [LQ17]. This also matches the algorithm for IQP2S which has an average-case running time of $q^{O(n)}$ [BFV13].

For practical algorithms, we draw experiences from the computational group theory community and the multivariate cryptography community. In the computational group theory community, the current status of the art is that one can hope to handle 10-tuples of alternating matrices of size $10 \times 10$ over $\mathbb{F}_{13}$, but absolutely not, for 3-tensors of local dimension say 100, even though in this case the input can still be stored in only a few megabytes.[3] In the multivariate cryptography community, the Gröbner basis technique [FP06] and certain combinatorial technique [BFV13] have been studied to tackle IQF2S problem. However, these techniques are not effective enough to break it [BFV13][4].

For quantum algorithms, 3TI seems difficult for the hidden subgroup approach, due to the reasons presented in Section 3.1.1.

Finally, let us also elaborate on the prospects of using those techniques for graph isomorphism [Bab16] and for isomorphism of quadratic polynomials with one secret [IQ19] to tackle 3TI. In general, the difficulties of applying these techniques seem inherent.

We first check out the graph isomorphism side. Recall that most algorithms for graph isomorphism, including Babai's [Bab16], are built on two families of techniques: group-theoretic, and combinatorial. To use the group-theoretic techniques, we need to work with matrix groups over finite fields instead of permutation groups. Algorithms for matrix groups over finite fields are in general far harder than those for permutation groups. For example, the basic membership problem is well-known to be solvable by Sims's algorithm [Sim78], while for matrix groups over finite fields of odd order, this was only recently shown to be efficiently solvable with a number-theoretic oracle and the algorithm is much more involved [BBS09]. To use the combinatorial techniques, we need to work with linear or multilinear structures instead of combinatorial structures. This shift poses severe limitations on the use of most combinatorial techniques, like individualizing a vertex. For example, it is quite expensive to enumerate all vectors in a vector space over a finite field, while this is legitimate to go over all elements in a set.

We then check out the isomorphism of quadratic polynomials with one secret side. The techniques for settling this problem as in [IQ19] are based on those developed for the module isomorphism problem [CIK97, BL08, IKS10], involutive algebras [Wil09], and computing algebra structures [FR85]. The starting point of that algorithm solves an easier problem, namely testing whether two matrix tuples are equivalent under the left-right multiplications. That problem is essentially linear, so the techniques for the module isomorphism problem can be used. After that we need to utilize the involutive algebra structure [Wil09] based on [FR85]. However, for 3TI, there is no such easier linear problem to start with, so it is not clear how those techniques can be applied.

To summarize, the 3-tensor isomorphism problem is difficult from all the four types of algorithms mentioned in Section 3.1. Furthermore, the techniques in the recent breakthrough on graph isomorphism [Bab16], and the solution of the isomorphism of quadratic polynomials with one secret [IQ19], seem not applicable to this problem. All these together support this problem as a strong candidate for the one-way assumption.

[3] We thank James B. Wilson, who maintains a suite of algorithms for $p$-group isomorphism testing, for communicating this insight to us from his hands-on experience. We of course maintain responsibility for any possible misunderstanding, or lack of knowledge regarding the performance of other implemented algorithms.

[4] In particular, as pointed out in [BFV13], one needs to be careful about certain claims and conjectures made in some literature on this research line.

### 3.3.3 Choices of the parameters

Having reviewed the current status of the tensor isomorphism problem, we lay out some principles of choosing the parameters for the security, namely the order $k$, the dimensions $d_i$, and the underlying field $\mathbb{F}$.

Let us first explain why we focus on $k = 3$, namely 3-tensors. Of course, $k$ needs to be $\geq 3$ as most problems about 2-tensors, i.e. matrices, are easy. We then note that there is certain evidence to support the possibility that the $k$-tensor isomorphism problem reduces to the 3-tensor isomorphism problem. That is, over certain fields, by [AS05, Theorem 5] and [FGS19], the degree-$k$ homogeneous form equivalence problem reduces to the the 3-tensor isomorphism problem in polynomial time. The former problem can be cast as an isomorphism problem for symmetric[5] $k$-tensors under a certain action of GL. From the practical viewpoint though, it will be interesting to investigate into the tradeoff between the local dimensions $d_i$ and $k$.

After fixing $k = 3$, it is suggested to set $d_1 = d_2 = d_3$. This is because of the argument when examining the worst-case time complexity in the above subsection.

Then for the underlying finite field $\mathbb{F}_q$, the intuition is that setting $q$ to be a large prime would be more secure. Note that we can still store an exponentially large prime using polynomially-many bits. This is because, if $q$ is small, then the "generic" behaviors as ensured by the Lang–Weil type theorems [LW54] may not be that generic. So some non-trivial properties may arise which then help with isomorphism testing. This is especially important for the pseudorandom assumption to be discussed Section 4. We then examine whether we want to set $q$ to be a large prime, or a large field with a small characteristic. The former one is preferred, because the current techniques in computer algebra and computational group theory, cf. [IQ19] and [BBS09], can usually work efficiently with large fields of small characteristics.

However, let us emphasize that even setting $q$ to be a constant, we do not have any concrete evidence for breaking GLAT as a one-way group action candidate. That is, the above discussion on the field size issue is rather hypothetical and conservative.

# 4 The pseudorandom action assumption

In this section, we introduce the new security assumption for group actions, namely pseudorandom group actions, which generalises the Decisional Diffie-Hellman assumption. In Section 5, we shall study the prospect of using the general linear action on tensors as a candidate for this assumption. Then in Section 7, we present the cryptographic uses of this assumption including signatures and pseudorandom functions.

**Definition 8.** Let $\mathcal{G}$ be a group family as specified before. Choose public parameters $\texttt{params} = (G, S, \alpha)$ to be $\mathcal{G}(1^\lambda)$. Sample $s \leftarrow S$ and $g \leftarrow G$. The group action pseudorandomness (GA-PR) problem is that given $(s, t)$, where $t = \alpha(g, s)$ or $t \leftarrow S$, decide which case $t$ is sampled from.

**Definition 9** (Pseudorandom group action game)**.** The pseudorandom group action game is the following game between a challenger and an adversary $\mathcal{A}$:

- The challenger and the adversary $\mathcal{A}$ agree on the public parameters $\texttt{params} = (G, S, \alpha)$ by choosing it to be $\mathcal{G}(1^\lambda)$ for some security parameter $\lambda$.
- Challenger samples random bit $b \in \{0, 1\}$, $s \leftarrow S$, $g \leftarrow G$, and chooses $t \leftarrow S$ if $b = 0$ and $t = g \cdot s$ if $b = 1$.

[5] A tensor $A = (A_{i_1,\ldots,i_k})$ is symmetric if for any permutation $\sigma \in \mathrm{S}_k$, and any index $(i_1, \ldots, i_k)$, $A_{i_1,\ldots,i_k} = A_{i_{\sigma(1)},\ldots,i_{\sigma(k)}}$.

- Give $(s,t)$ to $\mathcal{A}$ who produces a bit $a \in \{0,1\}$.
- We define the output of the game $\mathsf{GA\text{-}PR}_{\mathcal{A},\mathcal{G}}(1^\lambda) = 1$ and say $\mathcal{A}$ wins the game if $a = b$.

**Definition 10.** We say that the group-action pseudorandomness (GA-PR) problem is hard relative to $\mathcal{G}$, if for any polynomial-time quantum algorithm $\mathcal{A}$,

$$\Pr[\mathsf{GA\text{-}PR}_{\mathcal{A},\mathcal{G}}(1^\lambda) = 1] = \mathrm{negl}(\lambda).$$

Some remarks on this definition are due here.

**For transitive and almost transitive actions.** In the case of transitive group actions, as an easy corollary of Observation 1, we have the following.

**Observation 3.** GA-PR problem is hard, if the group action $\alpha$ is transitive.

Slightly generalizing the transitive case, it is not hard to see that GA-PR problem is hard, if there exists a "dominant" orbit $O \subseteq S$. Intuitively, this means that $O$ is too large such that random $s$ and $t$ from $S$ would both lie in $O$ with high probability. For example, consider the action of $\mathrm{GL}(n,\mathbb{F}) \times \mathrm{GL}(n,\mathbb{F})$ on $\mathrm{M}(n,\mathbb{F})$ by the left and right multiplications. The orbits are determined by the ranks of matrices in $\mathrm{M}(n,\mathbb{F})$, and the orbit of matrices of full-rank is dominant. But again, such group actions seems not very useful for cryptographic purposes. Indeed, we require the orbit structure to satisfy that random $s$ and $t$ do not fall into the same orbit. Let us formally put forward this condition.

**Definition 11.** We say that a group action $\alpha$ of $G$ on $S$ does not *have a dominant orbit*, if

$$\Pr_{s,t \leftarrow S}[s,t \text{ lie in the same orbit}] = \mathrm{negl}(\lambda).$$

**Assumption 2** (Pseudorandom group action (PRA) assumption)**.** There exists an $\mathcal{G}$ outputting a group action without a dominant orbit, relative to which the GA-PR problem is hard.

The name comes from the fact that the PRA assumption says 'in spirit' that the function $\Gamma : G \times S \to S \times S$ given by $\Gamma(g,s) = (g \cdot s, s)$ is a secure PRG. Here, it is only 'in spirit', because the PRA assumption does not include the usual expansion property of the PRG. Rather, it only includes the inexistence of a dominant orbit.

**Subsuming the classical Diffie-Hellman assumption.** We now formulate the classical decisional Diffie-Hellman (DDH) assumption as an instance of the pseudorandom group action assumption. To see this, we need the following definition.

**Definition 12.** Let $\alpha : G \times S \to S$ be a group action. The *d-diagonal action* of $\alpha$, denoted by $\alpha^{(d)}$, is the group action of $G$ on $S^d$, the Cartesian product of $d$ copies of $S$, where $g \in G$ sends $(s_1,\ldots,s_d) \in S^d$ to $(g \cdot s_1,\ldots,g \cdot s_d)$.

The following observation shows that the classical DDH can be obtained by instantiating GA-PR with a concrete group action.

**Observation 4.** Let $\alpha$ be the group action in Group Action 2. The classical Decisional Diffie-Hellman assumption is equivalent to the PRA assumption instantiated with $\alpha^{(2)}$, the 2-diagonal action of $\alpha$.

*Proof.* Recall from Group Action 2 defines an action $\alpha$ of $G \cong \mathbb{Z}_p^*$ on $S = C_p \setminus \{\mathrm{id}\}$ where $C_p$ is a cyclic group of order $p$. The 2-diagonal action $\alpha^{(2)}$ is defined by $a \in \mathbb{Z}_p^*$ sending $(s,t) \in S \times S$ to $(s^a, t^a)$. Note that while $\alpha$ is transitive, $\alpha^{(2)}$ is not, and in fact it does not have a dorminant orbit.

PRA instantiated with $\alpha^{(2)}$ then asks to distinguish between the following two distributions. The first distribution is $((s,t),(s',t'))$ where $s,t,s',t' \in_R S$. Since $\alpha$ is transitive, by Observation 1, this distribution is equivalent to $((s,s^a),(s^b,s^c))$, where $s \in_R S$ and $a,b,c \in_R G$. The second distribution is $((s,t),(s^b,t^b))$, where $s,t \in_R S$, and $b \in_R G$. Again, by Observation 1, this distribution is equivalent to $((s,s^a),(s^b,s^{ab}))$, where $s \in_R S$, and $a,b \in_R G$.

We then see that this is just the Decisional Diffie-Hellman assumption[6]. □

As will be explained in Section 5.1, the pseudorandom assumption is a strong one, in a sense much stronger than the one-way assumption. Therefore, Observation 4 is important because, by casting the classical Diffie-Hellman assumption as an instance of the pseudorandom assumption, it provides a non-trivial and well-studied group action candidate for this assumption.

Of course, the DDH assumption is no longer secure under quantum attacks. Recently, this assumption in the context of supersingular isogeny based cryptography has been proposed by De Feo and Galbraith in [FG19]. We will study the possibility for the 3-tensor isomorphism problem as a pseudorandom group action candidate in Section 5

**The $d$-diagonal pseudorandomness assumption.** Motivated by Observation 4, it will be convenient to specialize GA-PR to diagonal actions, and make the following assumption.

**Definition 13.** The $d$-diagonal pseudorandomness (GA-PR($d$)) problem for a group action $\alpha$, is defined to be the pseudorandomness problem for the $d$-diagonal group action $\alpha^{(d)}$.

We emphasize that GA-PR($d$) is just GA-PR applied to group actions of a particular form, so a special case of GA-PR. Correspondingly, we define PRA($d$) to be the assumption that GA-PR($d$) is hard relative to some $\mathcal{G}$.

Given a group action $\alpha : G \times S \to S$, let $F_\alpha = \{f_g : S \to S \mid g \in G, f_g(s) = g \cdot s\}$. It is not hard to see that PRA($d$) is equivalent to say that $F_\alpha$ is a $d$-query weak PRF in the sense of Maurer and Tessaro [MT08]. This gives a straightforward cryptographic use of the PRA($d$) assumption.

Given $d, e \in \mathbb{Z}^+$, $d < e$, it is clear that PRA($e$) is a stronger assumption than PRA($d$). Indeed, given an algorithm $A$ that distinguishes between

$$((s_1,\ldots,s_d),(g\cdot s_1,\ldots g\cdot s_d)) \text{ and } ((s_1,\ldots,s_d),(t_1,\ldots,t_d)),$$

where $s_i, t_j \leftarrow S$, and $g \leftarrow G$, one can use $A$ to distinguish between $((s_1,\ldots,s_e),(g\cdot s_1,\ldots g\cdot s_e))$ and $((s_1,\ldots,s_e),(t_1,\ldots,t_e))$, by just looking at the first $d$ components in each tuple.

The applications of the PRA assumption including more efficient quantum-secure digital signature schemes and pseudorandom function constructions are given in Section 7. Next, we will provide candidates to instantiate the PRA assumption.

[6]Here we use the version of DDH where the generator of the cyclic group is randomly chosen as also used in [CS98]. A recent discussion on distinction between fixed generators and random generators can be found in [BMZ19]

# 5 General linear actions on tensors: the pseudorandom action assumption

## 5.1 Requirements for a group action to be pseudorandom

Clearly, a first requirement for a group action to be pseudorandom is that it should be one-way. Further requirements naturally come from certain attacks. We have devised the following attack strategies. These attacks suggest that the pseudorandom assumption is closely related to the orbit closure intersection problem which has received considerable attention recently.

**Isomorphism testing in the average-case setting.** To start with, we consider the impact of an average-case isomorphism testing algorithm on the pseudorandom assumption. Recall that for a group action $\alpha : G \times S \to S$, an average-case algorithm is required to work for instances $(s, t)$ where $s \leftarrow S$ and $t$ is arbitrary. Let $n$ be the input size to this algorithm. The traditional requirement for an average-case algorithm is that it needs to work for *all but at most* $1/\operatorname{poly}(n)$ fraction of $s \in S$, like such algorithms for graph isomorphism [BES80] and for alternating matrix space isometry [LQ17]. However, in order for such an algorithm to break the pseudorandom assumption, it is enough that it works for a non-negligible, say $1/\operatorname{poly}(n)$, fraction of the instances. This is quite relaxed compared to the traditional requirement.

**The supergroup attack.** For a group action $\alpha : G \times S \to S$, a supergroup action of $\alpha$ is another group action $\beta : H \times S \to S$, such that (1) $G$ is a subgroup of $H$, (2) the restriction of $\beta$ to $G$, $\beta|_G$, is equal to $\alpha$. If it further holds that (3.1) the isomorphism problem for $H$ is easy, and (3.2) $\beta$ is not dominant, we will then have the following so-called *supergroup attack*. Give input $s, t \in S$, the adversary for the GA-PR problem of $\alpha$ will use the solver for the isomorphism problem for $H$ to check if $s, t$ are from the same orbit induced by $H$ and return 1 if they are from the same orbit and 0 otherwise. If $s, t$ are from the same orbit induced by $G$, the adversary always returns the correct answer as $G$ is a subgroup of $H$. In the case that $s, t$ are independently chosen from $S$, by the fact that $\beta$ is not dominant, the adversary will return the correct answer 0 with high probability.

**The isomorphism invariant attack.** Generalizing the condition (3) above, we can have the following more general strategy as follows. We now think of $G$ and $H$ as defining equivalence relations by their orbit structures. Let $\sim_G$ (resp $\sim_H$) be the equivalence relation defined by $G$ (resp. $H$). By the conditions (1) and (2), we have (a) $\sim_H$ is coarser than $\sim_G$. By the condition (3.1), we have (b) $\sim_H$ is easy to decide. By the condition (3.2), we have (c) $\sim_H$ have enough equivalence classes. Clearly, if a relation $\sim$, not necessarily defined by a supergroup $H$, satisfies (a), (b), and (c), then $\sim$ can also be used to break the PRA assumption for $G$.

Such an equivalence relation is more commonly known as an isomorphism invariant, namely those properties that are preserved under isomorphism. The sources of isomorphism invariants can be very versatile. The supergroup attack can be thought of as a special case of category where the equivalence relation is defined by being isomorphic under a supergroup action. Another somewhat surprising and rich "invariant" comes from geometry, as we describe now.

**The geometric attack.** In the case of matrix group actions, the underlying vector spaces usually come with certain geometry which can be exploited for the attack purpose. Let $\alpha$ be a group action of $G$ on $V \cong \mathbb{F}^d$. For an orbit $O \subseteq V$, let its Zariski closure be $\overline{O}$. Let $\sim$ be the equivalence relation on $V$, such that for $s, t \in O$, $s \sim t$ if and only if $\overline{O_s} \cap \overline{O_t} \neq \emptyset$. It is obvious that $\sim$ is a coarser

relation than $\sim_G$. Furthermore, except some degenerate settings when $m$ or $n$ are very small, there would be enough equivalence classes defined by $\sim$, because of the dimension reason. So (a) and (c) are satisfied. Therefore, if we could test efficiently whether the orbit closures of $s$ and $t$ intersect, (b) would be satisfied and we could break the PRA for $\alpha$. This problem, known as the orbit closure intersection problem, has received considerable attention recently.

Another straightforward approach based on this viewpoint is to recall that the geometry of orbit closures is determined by the ring of invariant polynomials [MFK94]. More specifically, the action of $G$ on $V$ induces an action on $\mathbb{F}[V]$, the ring of polynomial functions on $V$. As $V \cong \mathbb{F}^d$, $\mathbb{F}[V] \cong \mathbb{F}[x_1, \ldots, x_d]$. Those polynomials invariant under this induced action form a subring of $\mathbb{F}[V]$, denoted as $\mathbb{F}[V]^G$. If there exists one easy-to-compute, non-trivial, invariant polynomial $f$ from $\mathbb{F}[V]^G$, we could then use $f$ to evaluate on the input instances and distinguish between the random setting (where $f$ is likely to evaluate differently) and the pseudorandom setting (where $f$ always evaluates the same).

### 5.1.1 Example attacks

We now list some examples to illustrate the above attacks.

**An example of using the isomorphism invariant attack.** We first consider the isomorphism invariant attack in the graph isomorphism case. Clearly, the degree sequence, consisting of vertex degrees sorted from large to small, is an easy to compute isomorphism invariant. A brief thought suggests that this invariant is already enough to break the pseudorandom assumption for graph isomorphism.

**An example of using the geometric attack.** We consider a group action similar to the 3-tensor isomorphism case (Group Action 1), inspired by the quantum marginal problem [BFG$^+$18]. Given a 3-tensor of size $\ell \times n \times m$, we can "slice" this 3-tensor according to the third index to obtain a tuple of $m$ matrices of size $\ell$ by $n$. Consider the action of $G = \mathrm{O}(\ell, \mathbb{F}) \times \mathrm{O}(n, \mathbb{F}) \times \mathrm{SL}(m, \mathbb{F})$ on matrix tuples $\mathrm{M}(\ell \times n, \mathbb{F})^m$, where the three direct product factors act by left multiplication, right multiplication, and linear combination of the $m$ components, respectively. For a matrix tuple $(A_1, \ldots, A_m)$ where $A_i \in \mathrm{M}(\ell \times n, \mathbb{F})$, form an $\ell n \times m$ matrix $A$ where the $i$-th column of $A$ is obtained by straightening $A_i$ according to columns. Then $A^t A$ is an $m$ by $m$ matrix. The polynomial $f = \det(A^t A)$ is then a polynomial invariant for this action. For this note that the group $\mathrm{O}(\ell, \mathbb{F}) \times \mathrm{O}(n, \mathbb{F})$ can be embedded as a subgroup of $\mathrm{O}(\ell n, \mathbb{F})$, so its action becomes trivial on $A^t A$. Then the determinant is invariant under the $\mathrm{SL}(m, \mathbb{F})$. When $m < \ell n$, which is the interesting case, $\det(A^t A)$ is non-zero. It follows that we have a non-trivial, easy-to-compute, polynomial invariant which can break the PRA assumption for this group action.

**An example of using the supergroup attack.** We then explain how the supergroup attack invalidates the PRA($d$) assumption for certain families of group actions with $d > 1$.

Let $\alpha$ be a linear action of a group $G$ on a vector space $V \cong \mathbb{F}^N$. We show that as long as $d > N$, PRA($d$) does not hold. To see this, the action of $G$ on $V$ gives a homomorphism $\phi$ from $G$ to $\mathrm{GL}(V) \cong \mathrm{GL}(N, \mathbb{F})$. For any $g \in G$, and $v_1, \ldots, v_d \in V$, we can arrange an $N \times d$ matrix $S = [v_1, \ldots, v_d]$, such that $T = [\phi(g)v_1, \ldots, \phi(g)v_d] = \phi(g)[v_1, \ldots, v_d]$. On the other hand, for $u_1, \ldots, u_d \in V$, let $T' = [u_1, \ldots, u_d]$. Let us consider the row spans of $S$, $T$ and $T'$, which are subspaces of $\mathbb{F}^d$ of dimension $\leq N < d$. Clearly, the row spans of $S$ and $T$ are the same. On the other hand, when $u_i$'s are random vectors, the row span of $T'$ is unlikely to be the same as that of $S$. This gives an efficient approach to distinguish between $T$ and $T'$.

We can upgrade the above attack even further as follows. Let $\alpha$ be a linear action of $G$ on the linear space of matrices $M = \mathrm{M}(m \times n, \mathbb{F})$. Recall that $\mathrm{GL}(m, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F})$ acts on $M$ by left and right multiplications. Suppose $\alpha$ gives rise to a homomorphism $\phi : G \to \mathrm{GL}(m, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F})$. For $g \in G$, if $\phi(g) = (A, B) \in \mathrm{GL}(m, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F})$, we let $\phi_1(g) := A \in \mathrm{GL}(m, \mathbb{F})$, and $\phi_2(g) = B \in \mathrm{GL}(n, \mathbb{F})$. We now show that when $d > (m^2 + n^2)/(mn)$, PRA($d$) does not hold for $\alpha$. To see this, for any $g \in G$, and $S = (A_1, \ldots, A_d) \in \mathrm{M}(m \times n, \mathbb{F})^d$, let

$$T = (\phi_1(g)^t A_1 \phi_2(g), \ldots, \phi_1(g)^t A_d \phi_2(g)).$$

On the other hand, let $T' = (B_1, \ldots, B_d) \in M^d$. Since

$$\dim(S) = \dim(\mathrm{GL}(m \times n, \mathbb{F})^d) = mnd > m^2 + n^2 = \dim(\mathrm{GL}(m, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F})),$$

$\alpha$ does not have a dominant orbit (cf. Definition 11) This means that, when $B_i$'s are sampled randomly from $S$, $T'$ is unlikely to be in the same orbit as $S$. Now we use the fact that, the isomorphism problem for the action of $\mathrm{GL}(m, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F})$ on $S$ can be solved in deterministic polynomial time [IQ19, Proposition 3.2]. This gives an efficient approach to distinguish between $T$ and $T'$.

Note that the set up here captures the Group Actions 3 and 4 in Section 3.2.2. For example, suppose for Group Action 3, we consider linear codes which are $n/2$-dimensional subspaces of $\mathbb{F}_q^n$. Then we have $m = n/2$, so PRA(3) for this action does not hold, as $3 > (m^2 + n^2)/(mn) = 5/2$.

On the other hand, when $d \leq (m^2 + n^2)/(mn)$, such an attack may fail, simply because of the existence of a dominant orbit.

### 5.2 The general linear action on tensors as a pseudorandom action candidate

We have explained why the general linear action on tensors is a good candidate for the one-way assumption in Section 3. We now argue that, to the best of our knowledge, it is also a candidate for the pseudorandom assumption.

We have described the current status of average-case algorithms for 3-tensor isomorphism problem in Section 3.3.2. One may expect that, because of the relaxed requirement for the average-case setting as discussed in Section 5.1, the algorithms in [LQ17, BFV13] may be accelerated. However, this is not the case, because these algorithms inherently enumerate all vectors in $\mathbb{F}_q^n$, or improve somewhat by using the birthday paradox.

We can also explain why the relaxed requirement for the average-case setting is still very difficult, by drawing experiences from computational group theory, because of the relation between GLAT and Group Action 5, which in turn is closely related to the group isomorphism problem as explained in Section 3.2.2. In group theory, it is known that the number of non-isomorphic $p$-groups of class 2 and exponent $p$ of order $p^\ell$ is bounded as $p^{\frac{2}{27}\ell^3 + \Theta(\ell^2)}$ [BNV07]. The relaxed average-case requirement in this case then asks for an algorithm that could test isomorphism for a subclass of such groups containing non-isomorphic groups as many as $p^{\frac{2}{27}\ell^3 + \Theta(\ell^2)} / \operatorname{poly}(\ell, \log p) = p^{\frac{2}{27}\ell^3 + \Theta(\ell^2)}$. This is widely regarded as a formidable task in computational group theory: at present, we only know of a subclass of such groups with $p^{O(\ell^2)}$ many non-isomorphic groups that allows for an efficient isomorphism test [LW12].

The supergroup attack seems not useful here. The group $G = \mathrm{GL}(\ell, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F}) \times \mathrm{GL}(m, \mathbb{F})$ naturally lives in $\mathrm{GL}(\ell nm, \mathbb{F})$. However, by Aschbacher's classification of maximal subgroups of finite classical groups [Asc84], there are few natural supergroups of $G$ in $\mathrm{GL}(\ell nm, \mathbb{F})$. The obvious ones include subgroups isomorphic to $\mathrm{GL}(\ell n, \mathbb{F}) \times \mathrm{GL}(m, \mathbb{F})$, which is not useful because it has a dominant orbit (Definition 11).

The geometric attack seems not useful here either. The invariant ring here is trivial [DW00][7]. For the orbit closure intersection problem, despite some recent exciting progress in [BGdO$^+$18, BFG$^+$18, AGL$^+$18, DM18, IQS17], the current best algorithms for the corresponding orbit closure intersection problems still require exponential time.

Finally, for the most general isomorphism invariant attack, the celebrated paper of Hillar and Lim [HL13] is just titled "Most Tensor Problems Are NP-Hard." This suggests that getting one easy-to-compute and useful isomorphism invariant for GLAT is already a challenging task. Here, useful means that the invariant does not lead to an equivalence relation with a dominant class in the sense of Definition 11.

The above discussions not only provide evidence for GLAT to be pseudorandom, but also highlight how this problem connects to various mathematical and computational disciplines. We believe that this could serve a further motivation for all these works in various fields.

# 6 Primitives from the one-way assumption: proving quantum security

Based on any one-way group action, it is immediate to derive a one-way function family. A bit commitment also follows by standard techniques, which we shall discuss later on in Section 7.3.

In this section, we focus on the construction of a digital signature that exists in the literature. It follows a very successful approach of applying the Fiat-Shamir transformation on an identification scheme. However, proving quantum security of this generic method turns out to be extremely challenging and delicate. For this reason, we include a complete description of the construction and a formal security proof in the quantum setting.

## 6.1 Identification: definitions

An *identification scheme* ID consists of a triple of probabilistic polynomial-time algorithms[8] $(\mathrm{KG}, \mathcal{P}, \mathcal{V})$:

- Key generating: $(pk, sk) \leftarrow \mathrm{KG}(1^\lambda)$ generates a public key $pk$ and a secret key $sk$.
- Interactive protocol: $\mathcal{P}$ is given $(sk, pk)$, and $\mathcal{V}$ is only given $pk$. They then interact in a few rounds, and in the end, $\mathcal{V}$ outputs either 1 (i.e., "accept") or 0 (i.e., "reject").

We assume that the keys are drawn from some relation, i.e. $(sk, pk) \in R$, and $(\mathcal{P}, \mathcal{V})$ is an interactive protocol to prove this fact. Let

$$L_R := \{pk : \exists\, sk \text{ such that } (sk, pk) \in R\}\,,$$

be the set of valid public keys.

We will exclusively consider identification schemes of a special form, terms as $\Sigma$ protocols. In a $\Sigma$ protocol $(\mathcal{P}, \mathcal{V})$, three messages are exchanged in total:

- Prover's initial message: $I \leftarrow \mathcal{P}(sk, pk)$. $I$ is usually called the "commitment", and comes from $\{0,1\}^{\ell_{\mathrm{in}}}$.

[7] If instead of $\mathrm{GL}(\ell, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F}) \times \mathrm{GL}(m, \mathbb{F})$ we consider $\mathrm{SL}(\ell, \mathbb{F}) \times \mathrm{SL}(n, \mathbb{F}) \times \mathrm{SL}(m, \mathbb{F})$, the invariant ring is non-trivial – also known as the ring of semi-invariants for the corresponding GL action – but highly complicated. When $\ell = m = n$, we do not even know one single easy-to-compute non-trivial invariant. It further requires exponential degree to generate the whole invariant ring [DM19].

[8] We only consider classical protocols where the algorithms can be efficiently realized on classical computers.

- Verifier's challenge: $c \leftarrow \mathcal{V}(pk, I)$. Let the set of challenge messages (the challenge space) be $\{0,1\}^{\ell_{\mathrm{ch}}}$.
- Prover's response: prover computes a response $r \in \{0,1\}^{\ell_{\mathrm{re}}}$ based on $(sk, I, c)$ and its internal state.

Here $\ell_{\mathrm{in}}, \ell_{\mathrm{ch}}, \ell_{\mathrm{re}}$ are interpreted as functions of the security parameter $\lambda$. We omit writing $\lambda$ explicitly for the ease of reading.

A basic requirement of an ID is the *correctness*, which is basically the completeness condition of the interactive protocol $(\mathcal{P}, \mathcal{V})$ on correctly generated keys.

**Definition 14.** An $\mathrm{ID} = (\mathrm{KG}, \mathcal{P}, \mathcal{V})$ is correct if

$$\Pr\big[\mathcal{V}(pk) = 1 : (pk, sk) \leftarrow \mathrm{KG}(1^\lambda)\big] \geq 1 - \mathrm{negl}(\lambda).$$

Instead of defining security for ID directly, we usually talk about various properties of the $\Sigma$ protocol associated with ID, which will make the ID useful in various applications. In the following, we consider an adversary $\mathcal{A}$ which operates and shares states in multiple stages. Our notation does not explicitly show the sharing of states though.

**Definition 15.** (Adapting [Unr17, Definition 4]) Let $(\mathcal{P}, \mathcal{V})$ be a $\Sigma$ protocols with message length parameters $\ell_{\mathrm{in}}, \ell_{\mathrm{ch}}$ and $\ell_{\mathrm{re}}$.

- **Statistical soundness**: no adversary can generate a public key not in $L_R$ but manage to make $\mathcal{V}$ accept. More precisely, for any algorithm $\mathcal{A}$ (possibly unbounded),

$$\Pr\big[\mathcal{V}(pk, I, c, r) = 1 \wedge pk \notin L_R : r \leftarrow \mathcal{A}(pk, I, c), c \leftarrow \mathcal{V}(pk, I), (pk, I) \leftarrow \mathcal{A}(1^\lambda)\big] \leq \mathrm{negl}(\lambda).$$

- **Honest-verifier zero-knowledge (HVZK)**: there is a quantum polynomial-time algorithm $\mathcal{S}$ (the simulator) such that for any quantum polynomial-time adversary $\mathcal{A}$,

$$\Big|\Pr\big[\mathcal{A}(pk, I, c, r) = 1 : (I, c, r) \leftarrow (\mathcal{P}(sk), \mathcal{A}(pk)), (sk, pk) \leftarrow \mathrm{KG}(1^\lambda)\big] - \Pr\big[\mathcal{A}(pk, I, c, r) = 1 : (I, c, r) \leftarrow \mathcal{S}(pk), (sk, pk) \leftarrow \mathrm{KG}(1^\lambda)\big]\Big| \leq \mathrm{negl}(\lambda).$$

- **Computational special soundness**: from any two accepting transcripts with the same initial commitment, we can extract a valid secret key. Formally, there is a quantum polynomial-time algorithm $\mathcal{E}$ such that for any quantum polynomial-time $\mathcal{A}$, we have that

$$\Pr\big[(sk, pk) \notin R \wedge \mathcal{V}(pk, I, c, r) \wedge \mathcal{V}(pk, I, c', r') = 1 \wedge c \neq c' : (pk, I, c, r, c', r') \leftarrow \mathcal{A}(1^\lambda), sk \leftarrow \mathcal{E}(pk, I, c, r, c', r')\big] \leq \mathrm{negl}(\lambda).$$

- **Unique response**: it is infeasible to find two accepting responses for the same commitment and challenge. Namely, for any quantum polynomial-time $\mathcal{A}$,

$$\Pr\big[r \neq r' \wedge \mathcal{V}(pk, I, c, r) = 1 \wedge \mathcal{V}(pk, I, c, r') = 1 : (pk, I, c, r, r') \leftarrow \mathcal{A}(1^\lambda)\big] \leq \mathrm{negl}(\lambda).$$

- **Unpredictable commitment**: Prover's commitment has superlogarithmic collision-entropy.

$$\Pr\big[c = c' : c \leftarrow \mathcal{P}(sk, pk), c' \leftarrow \mathcal{P}(sk, pk), (sk, pk) \leftarrow \mathrm{KG}(1^\lambda)\big] \leq \mathrm{negl}(\lambda).$$

Protocol ID

Let $\mathcal{G}$ be a family of group actions that is OWA. Choose public parameters $\texttt{params} := (G, S, \alpha)$ to be $\mathcal{G}(1^\lambda)$. Construct ID $= (\mathrm{KG}_0, \mathcal{P}_0, \mathcal{V}_0)$ as follows:

- $\mathrm{KG}_0(1^\lambda)$: Sample uniformly random $s \leftarrow S$ and $g \leftarrow G$, and compute $t = \alpha(g, s)$. Output $pk := (s, t)$ and $sk := g$.
- $\mathcal{P}_0$ commitment: $\mathcal{P}_0$ picks a random $h \leftarrow G$. Compute $I = \alpha(h, s)$ and send to $\mathcal{V}_0$.
- $\mathcal{V}_0$ challenge: $c \leftarrow \{0, 1\}$.
- $\mathcal{P}_0$ response: if $c = 0$, $\mathcal{P}_0$ sends $r := h$; if $c = 1$, $\mathcal{P}_0$ sends $r := hg^{-1}$.
- $\mathcal{V}_0$ verdict: if $c = 0$, $\mathcal{V}_0$ outputs 1 iff. $\alpha(r, s) = I$; if $c = 1$, $\mathcal{V}_0$ outputs 1 iff. $\alpha(r, t) = I$.

Figure 1: Identification protocol ID based on GA-Inv.

### 6.2 Identification: construction from OWA

We construct an ID based on the GA-Inv problem. This is reminiscent of the famous zero-knowledge proof system for graph-isomorphism.

To get a statistically sound protocol, we compose ID's interactive protocol $(\mathcal{P}_0, \mathcal{V}_0)$ in parallel $\ell = \omega(\log \lambda)$ times. We denote the resulting protocol GA-ID.

Protocol GA-ID

Given ID $= (\mathrm{KG}_0, \mathcal{P}_0, \mathcal{V}_0)$. Choose public parameters $\texttt{params} := (G, S, \alpha)$ to be $\mathcal{G}(1^\lambda)$. Construct GA-ID $= (\mathrm{KG}, \mathcal{P}, \mathcal{V})$ as follows:

- $\mathrm{KG}(1^\lambda)$: run $\mathrm{KG}_0$ independently $\ell$ times. Output $pk := \{(s_i, t_i)\}_{i=1}^{\ell}$ and $sk := \{g_i\}_{i=1}^{\ell}$.
- $\mathcal{P}$ commitment: for $i = 1, \ldots, \ell$, run $\mathcal{P}_0$ to produce $h_i \leftarrow G$ and $I_i = \alpha(h_i, s_i)$. Each time $\mathcal{P}$ uses fresh randomness. Send $(I_i)_{i=1}^{\ell}$ to $\mathcal{V}$.
- $\mathcal{V}$ challenge: run $\mathcal{V}_0$ in parallel $\ell$ times, i.e., $c \leftarrow \{0, 1\}^\ell$.
- $\mathcal{P}$ response: for $i = 1, \ldots, \ell$, produce $r_i \leftarrow \mathcal{P}_0(c_i, sk, pk, h_i, I_i)$. Send $\mathcal{V}$ $(r_i)_{i=1}^{\ell}$.
- $\mathcal{V}$ verdict: for $i = 1, \ldots, \ell$, run $b_i \leftarrow \mathcal{V}_0(pk, I_i, c_i, r_i)$. Outputs 1 iff. all $b_i = 1$.

Figure 2: Identification protocol GA-ID

**Theorem 16.** GA-ID *has* correctness, statistically soundness, HVZK *and* unpredictable commitment*, assuming Assumption 1 holds.*

*Proof.* We prove the properties one by one.

- **Correctness**. This is clear.
- **Statistical soundness**. For any adversary $\mathcal{A}$ who produces some $pk = (s, t) \notin L_R$, it implies that for any $i \in [\ell]$, it can only answer one of the two challenges ($c_i = 0$ or 1) but not both. Since $c_i$'s are all uniformly chosen, $\mathcal{V}$ will reject except with probability $(\frac{1}{2})^\ell = \mathrm{negl}(\lambda)$ (noting that $\ell = \omega(\log \lambda)$).
- **HVZK**. We construct a simulator $\mathcal{S}$ in Fig. 3. The simulated transcript is identically distributed as the real execution.

Given $pk = \{(s_i, t_i)\}_{i=1}^{\ell}$, which is generated $(pk, sk) \leftarrow \mathrm{KG}(1^\lambda)$,

- $\mathcal{S}$ generates $c \leftarrow \{0,1\}^\ell$.
- For $i = 1, \dots \ell$, if $c_i = 0$, let $I_i := \alpha(h_i, s_i)$ and $r_i := h_i$; if $c_i = 1$, let $I_i := \alpha(h_i, t_i)$ and $r_i := h_i$. Here $h_i \leftarrow G$ are sampled randomly for all $i$.
- Output $(I, c, r) = (I_i, c_i, r_i)_{i=1}^{\ell}$.

Figure 3: Simulator $\mathcal{S}$

- **Special soundness**. If $\mathcal{A}$ can produce $(I, c, r)$ and $(I, c', r')$ that are both accepting with $c \neq c'$. Then at least $c_i \neq c'_i$ for some $i \in [\ell]$. The corresponding $r_i$ and $r'_i$ are hence $h$ and $hg^{-1}$ from which we can recover the secret key $g$.
- **Unpredictable commitment**. Two commitment messages collide only if $\alpha(g, s) = \alpha(g', s)$ for random $g, g' \leftarrow G$. All our candidate group actions are (almost) injective.

□

### 6.3 Digital signature from OWA

**Definition 17.** A *digital signature scheme* consists of a triple of probabilistic polynomial-time algorithms (KeyGen, Sign, Verify) where

- KeyGen: $(pk, sk) \leftarrow \mathrm{KeyGen}(1^\lambda)$ generates a pair of secret key and public key.
- Sign: on input $sk$ and message $m \in \mathcal{M}$, outputs $\sigma \leftarrow \mathrm{Sign}_{sk}(m)$.
- Verify: on input $pk$ and message-signature pair $(m, \sigma)$, output $\mathrm{Verify}_{pk}(m, \sigma) = \mathrm{acc/rej}$.

A signature is secure if no one without the secret key can forge a valid signature, even if it gets to see a number of valid message-signature pairs. This is modeled as giving an adversary the signing oracle. We give the formal definition below which explicitly incorporates a random oracle $H$ that is available to all users, and an adversary can access in quantum superposition. We stress that we do not allow quantum access to the signing oracle, which is a stronger attack model (cf. [BZ13]).

**Definition 18** (Unforgeability)**.** A signature scheme (KeyGen, Sign, Verify) is *unforgeable* iff. for all quantum polynomial-time algorithm $\mathcal{A}$,

$$\Pr\big[\mathrm{Verify}^H(pk, \sigma^*, m^*) = 1 \wedge m^* \notin \mathcal{L} : (pk, sk) \leftarrow \mathrm{KeyGen}(1^\lambda), (m^*, \sigma^*) \leftarrow \mathcal{A}^{H, \mathrm{Sign}_{sk}}(\lambda, pk)\big] \leq \mathrm{negl}(\lambda).$$

Here $\mathcal{L}$ contains the list of messages that $\mathcal{A}$ queries to the (classical) signing oracle $\mathrm{Sign}_{sk}(\cdot)$.

Note the unforgeability does not rule out an adversary that produces a new signature on some message that it has queried before. Strong unforgeability takes this into account.

**Definition 19** (Strong Unforgeability)**.** A signature scheme $(\textsc{KeyGen}, \textsc{Sign}, \textsc{Verify})$ is *strongly unforgeable* iff. for all quantum polynomial-time algorithm $\mathcal{A}$,

$$\Pr\big[\textsc{Verify}^H(pk, \sigma^*, m^*) = 1 \wedge (m^*, \sigma^*) \in \mathcal{L} : (pk, sk) \leftarrow \textsc{KeyGen}(1^\lambda), (m^*, \sigma^*) \leftarrow \mathcal{A}^{H, \textsc{Sign}_{sk}}(\lambda, pk)\big] \leq \mathrm{negl}(\lambda).$$

Here $\mathcal{L}$ contains the list of message *and signature* pairs that $\mathcal{A}$ queries to the (classical) signing oracle $\textsc{Sign}_{sk}(\cdot)$.

Fiat and Shamir proposed a simple, efficient, and generic method that converts an identification scheme to a signature scheme using a hash function, and the security can be proven in the random oracle model [FS86, PS00]. Relatively recent, Fischlin proposed a variant to partly reduce the reliance on the random oracle [Fis05]. However, as shown in [ARU14], both of them seem difficult to admit a security proof in the quantum setting. Instead, Unruh [Unr15] proposed a less efficient transformation, hereafter referred to as Unruh transformation, and proved its security in the quantum random oracle model. Our GA-ID satisfies the conditions required in Unruh transformation, and hence we can apply it and obtain a digital signature scheme GA-SIGN.

**Theorem 20** (Adapting Corollary 19 & Theorem 23 & of [Unr15])**.** *If an identification scheme* ID *have correctness, HVZK and special soundness, then protocol* SIGN *in Fig. 4 is a strongly unforgeable signature in* QRO*.*

Since GA-ID has these properties, we instantiate SIGN with GA-ID and call the resulting signature scheme GA-SIGN.

**Corollary 21.** *If Assumption 1 holds,* GA-SIGN *is a strongly unforgeable signature in* QRO*.*

# 7 Quantum-secure primitives from the pseudorandom action assumption

## 7.1 Improved Digital signature based on PRA via Fiat-Shamir

In this subsection, we show that if we accept the possibly stronger assumption of PRA, we can apply the standard Fiat-Shamir transform to GA-ID, and obtain a more efficient signature scheme in QRO. This is due to a recent work by Unruh [Unr17][9], where he shows that if one can equip ID with a "dual-mode" key generation, Fiat-Shamir will indeed work in QRO. A dual-mode key is a fake public key $\widetilde{pk}$ that is indistinguishable from a valid public key. Nonetheless, $\widetilde{pk}$ has no corresponding secret key (i.e., $\widetilde{pk} \notin L_R$).

**Definition 22** (Dual-mode key generator, adapting [Unr17])**.** An algorithm KG is a *dual-mode key generator* for a relation $R$ iff.

- KG is quantum polynomial-time,
- $\Pr\big[(sk, pk) \in R : (sk, pk) \leftarrow \mathrm{KG}(1^\lambda)\big] \geq 1 - \mathrm{negl}(\lambda)$.

[9] [KLS18] includes a similar result, which is primarily tailored to lattice-based identification schemes.

Signature scheme SIGN based on Unruh transformation

Let $\lambda$ be the security parameter. Let $t$ and $s$ be integers such that $t \log s = \omega(\log \lambda)$. Let $\ell_{\text{in}}, \ell_{\text{ch}}, \ell_{\text{re}}$ be the length of the commitment, challenge and response respectively. Choose hash functions $H_1 : \{0,1\}^* \to \{1,\dots,s\}^t$ and $H_2 : \{0,1\}^{\ell_{\text{re}}} \to \{0,1\}^{\ell_{\text{re}}}$. For an identification scheme ID, we construct a signature scheme SIGN as follows:

- KeyGen: run $\mathrm{KG}(1^\lambda)$ of ID to obtain $(pk, sk)$.
- Sign: for input $sk$ and message $m$ do the following
  i) for $i = 1,\dots,t$ and $j = 1,\dots,s$, generate $I_i \leftarrow \mathcal{P}(pk, sk)$, $c_{i,j} \leftarrow \{0,1\}^{\ell_{\text{ch}}} \backslash \{c_{i,1},\dots,c_{i,j-1}\}$. Then compute $r_{i,j} \leftarrow \mathcal{P}(c_{i,j}, sk)$ and $h_{i,j} := H_2(r_{i,j})$.
  ii) compute $H_1(pk, m, (I_i)_{i=1,\dots,t}, (c_{i,j}, h_{i,j})_{i=1,\dots,t,j=1,\dots,s})$. Partition the output as $J_1 \| \dots \| J_t$ where each $J_i \in [s]$.
  iii) output $\sigma := ((I_i)_{i=1,\dots,t}, (c_{i,j}, h_{i,j})_{i=1\dots,t,j=1,\dots,s}, (r_{i,J_i})_{i=1,\dots,t})$.
- Verify: for input $(m, \sigma)$ and public key $pk$,
  i) parse $\sigma$ as $((I_i)_{i=1,\dots,t}, (c_{i,j}, h_{i,j})_{i=1,\dots,t,j=1,\dots,s}, (r_i)_{i=1,\dots,t})$.
  ii) compute $J_1 \| \dots \| J_t := H_1(pk, m, (I_i)_{i=1,\dots,t}, (c_{i,j}, h_{i,j})_{i=1,\dots,t,j=1,\dots,s})$.
  iii) for $i = 1,\dots,t$, check $c_{i,1},\dots,c_{i,s}$ distinct; check $\mathcal{V}(pk, I_i, c_{i,J_i}, r_i) = 1$; check $h_{i,J_i} = H_2(r_i)$.
  iv) accept if all checks pass.

Figure 4: Unruh transformation

- for all quantum polynomial-time algorithm $\mathcal{A}$, there is a quantum polynomial-time algorithm $\mathrm{KG}^*$ such that

$$\left|\Pr\left[\mathcal{A}(pk) = 1 : (sk, pk) \leftarrow \mathrm{KG}(1^\lambda)\right] - \Pr\left[\mathcal{A}(\widetilde{pk}) = 1 : \widetilde{pk} \leftarrow \mathrm{KG}^*(1^\lambda)\right]\right| \leq \mathrm{negl}(\lambda),$$

and

$$\Pr\left[\widetilde{pk} \in L_R : \widetilde{pk} \leftarrow \mathrm{KG}^*(1^\lambda)\right] \leq \mathrm{negl}(\lambda).$$

**Theorem 23.** KG *in* GA-ID *is a dual-mode key generator, if Assumption 2 holds.*

*Proof.* We construct $\mathrm{KG}^*$ as follows:

1. choose $(G, S, \alpha)$ to be $\mathcal{G}(1^\lambda)$;
2. sample $s, t \leftarrow S$ uniformly;
3. output $\widetilde{pk} := (s, t)$.

By Assumption 2, it follows that

$$\left|\Pr\left[\mathcal{A}(pk) = 1 : (sk, pk) \leftarrow \mathrm{KG}(1^\lambda)\right] - \Pr\left[\mathcal{A}(\widetilde{pk}) = 1 : \widetilde{pk} \leftarrow \mathrm{KG}^*(1^\lambda)\right]\right| \leq \mathrm{negl}(\lambda)\,.$$

In addition,

$$\Pr\big[\, pk \in L_R : pk \leftarrow \mathrm{KG}^*(1^\lambda) \,\big] = \frac{|G|}{|S|} \leq \mathrm{negl}(\lambda). \qquad \square$$

Fiat-Shamir transformation

Let $\lambda$ be the security parameter, $\ell_{\mathrm{in}}, \ell_{\mathrm{ch}}, \ell_{\mathrm{re}}$ be the length of the commitment, challenge and response respectively. Choose a hash function $H : \{0,1\}^* \to \{0,1\}^{\ell_{\mathrm{re}}}$. Given an identification scheme ID, we construct a signature scheme FS-SIGN as follows:

- KeyGen: run $(sk, pk) \leftarrow \mathrm{KG}(1^\lambda)$. Output $(sk, pk)$.
- Sign: for input message $m$ and $sk$,
  i) compute $I \leftarrow \mathcal{P}(sk, pk)$, $c = H(pk, m, I)$, and $r \leftarrow \mathcal{P}(sk, pk, I, c)$.
  ii) output $\sigma := I \| r$.
- Verify: for input $pk$ and $(m, \sigma)$,
  i) parse $\sigma$ as $I \| r$.
  ii) compute $c := H(pk, m, I)$.
  iii) accept if $\mathcal{V}(I, c, r) = 1$.

Figure 5: Fiat-Shamir transformation

**Theorem 24.** *(Adapting [Unr17, Corollary 33]) If an identification scheme* ID *has correctness, HVZK, statistical soundness, unpredictable commitments,* $\ell_{\mathrm{ch}}$ *is superlogarithmic, and* KG *is a dual-mode key generator. Then in* QRO*,* FS-SIGN *obtained from Fiat-Shamir transform (Construction in Fig. 5) is weakly unforgeable. If* ID *has unique responses, the signature scheme is strongly unforgeable.*

Since GA-ID has all these properties, we can instantiate Construction FS-SIGN with GA-ID. Call the resulting signature scheme GA-FS-SIGN.

**Corollary 25.** GA-FS-SIGN *is strongly unforgeable, if Assumption 2 holds.*

Note that GA-FS-SIGN is much more efficient than GA-SIGN. In particular, GA-FS-SIGN only invokes the underlying $(\mathcal{P}, \mathcal{V})$ once as opposed to superpolylogarithmic times in GA-SIGN.

### 7.2 Quantum-secure pseudorandom functions based on PRA

Finally, we discuss how to construct quantum-secure pseudorandom functions using the PRA assumption. Basically, we will show that we can instantiate the GGM construction [GGM86] using the PRA assumption. To do this, we need to first discuss constructing pseudorandom generators.

**(Keyed) pseudorandom generators.** We already have mentioned that we can construct a PRG $\Gamma : S \times G \to S \times S$, given by

$$\Gamma(s, g) := (s, g \cdot s).$$

In fact, we may modify this construction slightly to obtain a form of PRG with much better stretching almost for free as follows. For $s \in S$, we define $\Gamma_s : G \to S$ by

$$\Gamma_s(g) := g \cdot s.$$

This can be considered as a 'keyed PRG', where $s$ is a public key for the PRG instance $\Gamma_s$, and this instance stretches the seed $g \leftarrow G$ to $g \cdot s$. Such notion of a keyed PRG is informally given in [HLY12], but surely this notion was used implicitly in many works previously. We may give a formal definition of this notion as follows.

**Definition 26** (Keyed PRG)**.** A *keyed pseudorandom generator*, or a keyed PRG, is a pair of probabilistic polynomial-time algorithms $(\mathrm{KG}, \mathrm{PRG})$:

- Key generator: $k \leftarrow \mathrm{KG}(1^\lambda)$ generates a *public* key $k \in \mathcal{K}$ describing an instance of the keyed PRG.
- Pseudorandom generator: given $k$ sampled by $\mathrm{KG}(1^\lambda)$, $\mathrm{PRG}_k : \mathcal{X} \to \mathcal{Y}$ stretches a uniform element $x \leftarrow \mathcal{X}$ to produce an element $\mathrm{PRG}_k(x) \in \mathcal{Y}$. Note that this PRG algorithm is required to be deterministic.

In the above, $\mathcal{K}$ is the key space of the keyed PRG, and $\mathcal{X}$, $\mathcal{Y}$ are the domain and the codomain of the keyed PRG, respectively. They are implicitly parametrized by the main parameter $\lambda$. Also, it is required that $|\mathcal{Y}| > |\mathcal{X}|$.

**Definition 27** (Security of a keyed PRG)**.** We say that a keyed PRG, $\Gamma = (\mathrm{KG}, \mathrm{PRG})$, is *secure*, if for any quantum polynomial-time adversary $\mathcal{A}$, we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{prg}}_{\Gamma}(\mathcal{A}) := \Big|&\Pr\big[\mathcal{A}(k, \mathrm{PRG}_k(x)) = 1 \,:\, x \leftarrow \mathcal{X}, k \leftarrow \mathrm{KG}(1^\lambda)\big] \\ &- \Pr\big[\mathcal{A}(k, y) = 1 \,:\, y \leftarrow \mathcal{Y}, k \leftarrow \mathrm{KG}(1^\lambda)\big]\Big| \le \mathrm{negl}(\lambda).\end{aligned} \tag{1}$$

Again, it is immediate that PRA assumption implies that $g \mapsto g \cdot s$ is a secure keyed PRG, where $s$ is the key and $g$ is the seed.

**Doubling keyed PRGs.** The keyed PRG $\Gamma_s$ that we have described above is of form $\Gamma_s : G \to S$. While $|S| \gg |G|$, having $S$ which might 'look different' from $G$ can be inconvenient for some applications, for example, constructing a PRF via the GGM construction. So, here we would like to construct a 'doubling' keyed PRG out of the previous construction, using randomness extraction.

The idea is simple: $\Gamma_s(g)$ would look uniform random over $S$ for average $s$, so we can use a randomness extractor to produce a pseudorandom bit string of enough length, and use that to sample two group elements of $G$. Overall, the construction would be of form $G \to G \times G$, while the PRF key would include not only the point $s \in S$ but also the random seed for the randomness extraction. For concreteness, we may use the Leftover Hash Lemma (LHL) [HILL99], but in fact any strong randomness extractor would be all right.

More concretely, let $R_G : \{0,1\}^p \to G$ be the sampling algorithm for the group $G$ which samples a random element of $G$, (statistically close to) uniform. Note that this $R_G$ is required for our group $G$. In fact, Babai [Bab91] gives an efficient Monte Carlo algorithm for sampling a group element of a finite group in a very general setting which is applicable to all of our instantiations.

Let $\mathcal{H} = \{h : S \to \{0,1\}^r\}$ be a family of 2-universal hash functions, where $r$ is sufficiently smaller than $\log |S|$. LHL implies, informally, that $(h, h(s))$ and $(h, u)$ are statistically indistinguishable, when $h \leftarrow \mathcal{H}$, $s \leftarrow S$, $u \leftarrow \{0,1\}^r$ are uniform and independent. Let us assume $\log |S|$ is large enough so that we can take $r = 2p$.

Then, we may construct a doubling keyed PRG $(\mathrm{KG}, \mathrm{PRG})$ as follows:

- Choose public parameters $\texttt{params}(G, S, \alpha)$ to be $\mathcal{G}(1^\lambda)$.
- Key generator: $\mathrm{KG}(1^\lambda)$ samples $s \leftarrow S$, $h \leftarrow \mathcal{H}$, and outputs $k := (s, h)$.
- Pseudorandom generator: $\mathrm{PRG}_k(g) := (R_G(r_0), R_G(r_1))$, where $r_0$ and $r_1$ are the left half and the right half of $h(g \cdot s)$, respectively.

In short, this keyed PRG stretches the seed $g \leftarrow G$ to $g \cdot s$, and extracts a pseudorandom bit string of length $2p$, and use that to sample two independent-looking group elements. The security of this construction comes from the PRA assumption and the Leftover Hash Lemma.

**Pseudorandom functions.** Of course, the notion of a PRF [GGM86] is well-known and well-established. Here, following Maurer and Tessaro [MT08], we are going to extend the notion of PRF somewhat so that it may also have an extra 'public key' part.

**Definition 28** (Pseudorandom function)**.** A pseudorandom function (PRF) is a polynomial-time computable function $f$ of form $f : \mathcal{P} \times \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. We call the sets $\mathcal{P}$, $\mathcal{K}$, $\mathcal{X}$, $\mathcal{Y}$ as the public-key space, the key space, the domain, and the codomain of $f$, respectively.

We would often write $f(p, k, x)$ as $f_p(k, x)$.

Note that we may regard an 'ordinary' PRF as a special case of above where it has a trivial, empty public key.

In this paper, we consider quantum-secure PRFs [Zha12], or, sometimes called QPRFs, whose security is defined as follows.

**Definition 29** (Security of a PRF)**.** Let $f : \mathcal{P} \times \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a PRF. We say that $f$ is *quantum-secure*, if for any quantum polynomial-time adversary $\mathcal{A}$ which can make quantum superposition queries to its oracle, we have the following:

$$\mathbf{Adv}_f^{\mathsf{prf}}(\mathcal{A}) := \left| \Pr\left[\mathcal{A}^{f_p(k,\cdot)}(p) = 1\right] - \Pr\left[\mathcal{A}^{\rho}(p) = 1\right] \right| = \mathrm{negl}(\lambda),$$

where $p \leftarrow \mathcal{P}$, $k \leftarrow \mathcal{K}$, $\rho \leftarrow \mathcal{Y}^{\mathcal{X}}$ are uniformly and independently random and $\lambda$ is the security parameter.

Suppose we have a secure, doubling keyed PRG $\Gamma = (\mathrm{KG}, \mathrm{PRG})$ where $\mathrm{PRG}_s$ is of form

$$\mathrm{PRG}_s : \mathcal{K} \to \mathcal{K} \times \mathcal{K}.$$

Writing the first component and the second component of $\mathrm{PRG}_s(k)$ as $f_s(k, 0)$ and $f_s(k, 1)$, we obtain a PRF $f$ of form

$$f : \mathcal{S} \times \mathcal{K} \times \{0, 1\} \to \mathcal{K}.$$

Here, $\mathcal{S}$ is the public-key space of $f$, which is the key space of the keyed PRG $\Gamma$. The key space of $f$ is $\mathcal{K}$, and the domain and the codomain of $f$ are $\{0, 1\}$ and $\mathcal{K}$, respectively.

Moreover, we can immediately see that the security of the one-bit PRF $f$ is exactly equivalent to the security of $\Gamma$ as a keyed PRG. In fact, we can say that the security of $f$ is just a re-statement of the security of $\Gamma$.

Now we may apply the GGM construction to $f$ to define the following PRF $\mathrm{GGM}[f] : \mathcal{S} \times \mathcal{K} \times \{0, 1\}^l \to \mathcal{K}$, where

$$\mathrm{GGM}[f]_s(k, x_1 \dots x_l) := f_s(\dots f_s(f_s(k, x_1), x_2), \dots, x_l).$$

In fact, the above is the same as the cascade construction for the one-bit PRF $f$.

When we instantiate the GGM construction using an ordinary PRG, or when we instantiate the cascade construction using an ordinary PRF (without the public-key part), the quantum security is already established [Zha12, SY17]. The only difference is that here we instantiate the construction using a keyed PRG, or, equivalently, a one-bit PRF with a public key.

Following [Zha12] or [SY17], we can define a version of oracle security for such a PRF with a public key.

**Definition 30** (Oracle security of a PRF)**.** Let $f : \mathcal{P} \times \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a PRF. We say that $f$ is *oracle-secure with respect to an index set* $\mathcal{I}$, if for any quantum polynomial-time adversary $\mathcal{A}$ which can make quantum superposition queries to its oracle, we have the following:

$$\mathbf{Adv}^{\mathsf{os\text{-}prf}}_{f,\mathcal{I}}(\mathcal{A}) := \left|\Pr\left[\mathcal{A}^{O_0}(p) = 1\right] - \Pr\left[\mathcal{A}^{O_1}(p) = 1\right]\right| = \mathrm{negl}(\lambda),$$

where the oracles $O_0, O_1$ are defined as

$$O_0(i, x) := f_p(\kappa(i), x), \quad O_1(i, x) := \rho(i, x),$$

and $p \leftarrow \mathcal{P}$, $\kappa \leftarrow \mathcal{K}^{\mathcal{I}}$, $\rho \leftarrow \mathcal{Y}^{\mathcal{I}\times\mathcal{X}}$ are chosen uniform randomly and independently.

And, as in [SY17], we show that if a PRF with a public key is secure, then it is also oracle-secure.

**Theorem 31.** *Let $f : \mathcal{P} \times \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a PRF. Suppose that it is secure as a PRF. Then, it is also oracle-secure.*

*Proof.* Here is a brief sketch of the proof. We are going to use the notion of relative (oracle) indistinguishability introduced in [SY17]. Our random oracle $H$ would be a very simple one, $H : \{*\} \to \mathcal{P}$, where $\{*\}$ is the singleton set containing only one element. Given this $p = H(*) \leftarrow \mathcal{P}$, we define two distributions $D_0, D_1$ of functions of form $\mathcal{X} \to \mathcal{Y}$.

- $D_0$: to sample a function $g$ from $D_0$, sample $k \leftarrow \mathcal{K}$, and define

$$g(x) := f_p(k, x).$$

- $D_1$: to sample a function $g$ from $D_1$, simply sample a uniform random function $g : \mathcal{X} \to \mathcal{Y}$.

Then, the security of $f$ is in fact equivalent to indistinguishability of $D_0$ and $D_1$ relative to the simple oracle $H$. Again according to [SY17], when two function distributions are indistinguishable relative to $H$, then they are oracle-indistinguishable relative to $H$. We can also observe that this is equivalent to the oracle security of $f$ defined as above. □

Finally, the security of the GGM construction comes from the oracle security.

**Theorem 32.** *Suppose that $f : \mathcal{S} \times \mathcal{K} \times \{0, 1\} \to \mathcal{K}$ is a secure PRF. Then, the GGM construction* $\mathrm{GGM}[f] : \mathcal{S} \times \mathcal{K} \times \{0, 1\}^l \to \mathcal{K}$ *is also secure.*

*Proof.* The proof is essentially identical to that of Zhandry [Zha12] or Song and Yun [SY17]; since $f$ is secure as a PRF, it is also oracle-secure. This allows the same hybrid argument in the security proof for GGM in [Zha12], or the security proof for the cascade construction in [SY17]. □

(Perfectly hiding and computationally binding) Bit commitment

Suppose Alice wants to commit to Bob a bit $b \in \{0,1\}$.

1. Bob samples $s_0 \in S$ and $g \in G$, and computes $s_1 = g \cdot s_0$.
2. Bob convinces Alice that $s_0$ and $s_1$ are in the same orbit, using the identification protocol in Section 6.1.
3. To commit to $b \in \{0,1\}$, Alice samples $h \in G$, computes $t = h \cdot s_b$, and sends $t$.
4. To open $b \in \{0,1\}$, Alice sends $h$ to Bob, and Bob verifies that $h \cdot s_b = t$.

Figure 6: A bit commitment scheme based on OWA

### 7.3 Bit commitment schemes based on OWA and PRA

Based on OWA (Assumption 1), Brassard and Yung [BY90] describe a bit commitment scheme, which we can easily adapt and instantiate it with non-abelian group actions.

We briefly argue how the security conditions are met. A formal proof (against both classical and quantum attacks) can be obtained along the same line. Let $b' = 1 - b$.

- *Binding*[10]: in order to change her mind, Alice needs to compute $h'$ such that $h' \cdot s_{b'} = t$. If she can do that, given that she already knows $h \cdot s_b = t$, she can compute $g$. This violates the one-way assumption.
- *Hiding*: from Bob's viewpoint, since $s_0$ and $s_1$ are in the same orbit, whichever bit Alice commits, the distributions of $t$ are the same.

Let us then examine an alternative route to bit commitment based on the PRA assumption. Consider the function below:

$$\begin{aligned} T : S \times G &\to S \times S \\ (s, g) &\mapsto (s, g \cdot s)\,. \end{aligned}$$

It is easy to see that this gives a quantum-secure *pseudorandom generator* (PRG) based upon Assumption 2, assuming that the size $|S|$ is larger than $|G|$. Therefore, after we apply the Blum-Micali amplification to increase the expansion to triple, we can plug it into Naor's commitment [Nao91] and get a *quantum computationally hiding* and *statistically binding* commitment.

**Theorem 33.** *There is a* perfectly hiding *and* computationally binding *bit commitment, if Assumption 1 holds; there is a* quantum computationally hiding *and* statistically binding *commitment scheme, if Assumption 2 holds.*

**Acknowledgement.** Y.Q. would like to thank Joshua A. Grochow for explaining the results in [FGS19] to him. Y.Q. was partially supported by Australian Research Council DE150100720. F.S. was partially supported by the U.S. National Science Foundation under CCF-1816869. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the

[10] Here we do not consider more sophisticated binding notions in the quantum setting (Cf. [Unr16]).

author(s) and do not necessarily reflect the views of the National Science Foundation. A.Y. was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2016-6-00598, The mathematical structure of functional encryption and its analysis).

## References

[AGL+18] Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Mendes de Oliveira, and Avi Wigderson. Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 172–181, 2018.

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *FOCS 2014*, pages 474–483, 2014.

[AS05] Manindra Agrawal and Nitin Saxena. Automorphisms of finite rings and applications to complexity of problems. In *STACS 2005, 22nd Annual Symposium on Theoretical Aspects of Computer Science, Stuttgart, Germany, February 24-26, 2005, Proceedings*, pages 1–17, 2005.

[Asc84] Michael Aschbacher. On the maximal subgroups of the finite classical groups. *Inventiones mathematicae*, 76(3):469–514, 1984.

[Bab79] László Babai. Monte-Carlo algorithms in graph isomorphism testing. Technical Report 79-10, Dép. Math. et Stat., Université de Montréal, 1979.

[Bab91] László Babai. Local expansion of vertex-transitive graphs and random generation in finite groups. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 164–174, 1991.

[Bab16] László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 684–697, 2016.

[BBD09] Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. *Post-quantum cryptography*. Springer, 2009.

[BBS09] László Babai, Robert Beals, and Ákos Seress. Polynomial-time theory of matrix groups. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 55–64, 2009.

[BC86] Gilles Brassard and Claude Crépeau. Non-transitive transfer of confidence: A perfect zero-knowledge interactive protocol for SAT and beyond. In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 188–195, 1986.

[BES80] László Babai, Paul Erdös, and Stanley M. Selkow. Random graph isomorphism. *SIAM J. Comput.*, 9(3):628–635, 1980.

[BFFP11] Charles Bouillaguet, Jean-Charles Faugère, Pierre-Alain Fouque, and Ludovic Perret. Practical cryptanalysis of the identification scheme based on the isomorphism of polynomial with one secret problem. In *International Workshop on Public Key Cryptography*, pages 473–493. Springer, 2011.

[BFG+18] Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Mendes de Oliveira, Michael Walter, and Avi Wigderson. Efficient algorithms for tensor scaling, quantum marginals, and moment polytopes. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 883–897, 2018.

[BFP15] Jérémy Berthomieu, Jean-Charles Faugère, and Ludovic Perret. Polynomial-time algorithms for quadratic isomorphism of polynomials: The regular case. *J. Complexity*, 31(4):590–616, 2015.

[BFV13] Charles Bouillaguet, Pierre-Alain Fouque, and Amandine Véber. Graph-theoretic algorithms for the "isomorphism of polynomials" problem. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 211–227, 2013.

[BGdO+18] Peter Bürgisser, Ankit Garg, Rafael Mendes de Oliveira, Michael Walter, and Avi Wigderson. Alternating minimization, scaling algorithms, and the null-cone problem from invariant theory. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 24:1–24:20, 2018.

[BL08] Peter A. Brooksbank and Eugene M. Luks. Testing isomorphism of modules. *Journal of Algebra*, 320(11):4020 – 4029, 2008.

[BMZ19] James Bartusek, Fermi Ma, and Mark Zhandry. The distinction between fixed and random generators in group-based assumptions. *IACR Cryptology ePrint Archive*, 2019:202, 2019.

[BNV07] Simon R. Blackburn, Peter M. Neumann, and Geetha Venkataraman. *Enumeration of finite groups*. Cambridge Univ. Press, 2007.

[Bon98] Dan Boneh. The decision Diffie-Hellman problem. In *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, pages 48–63, 1998.

[BS84] László Babai and Endre Szemerédi. On the complexity of matrix group problems I. In *25th Annual Symposium on Foundations of Computer Science, West Palm Beach, Florida, USA, 24-26 October 1984*, pages 229–240, 1984.

[BY90] Gilles Brassard and Moti Yung. One-way group actions. In *Advances in Cryptology - CRYPTO 1990*, pages 94–107, 1990.

[BZ13] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In *Advances of Cryptology – Crypto 2013*, volume 8043 of *LNCS*, pages 361–379, 2013.

[Che16] Lily Chen. *Report on post-quantum cryptography*. NIST.GOV, 2016.

[CIK97] Alexander Chistov, Gábor Ivanyos, and Marek Karpinski. Polynomial time algorithms for modules over finite dimensional algebras. In *Proceedings of the 1997 international symposium on Symbolic and algebraic computation*, ISSAC '97, pages 68–74, New York, NY, USA, 1997. ACM.

[CJS14] Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014.

[Cou06] Jean Marc Couveignes. Hard homogeneous spaces. *IACR Cryptology ePrint Archive*, 2006:291, 2006.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25. Springer, 1998.

[CvD10] Andrew M. Childs and Wim van Dam. Quantum algorithms for algebraic problems. *Rev. Mod. Phys.*, 82:1–52, Jan 2010.

[DFJP14] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[DM18] Harm Derksen and Visu Makam. Algorithms for orbit closure separation for invariants and semi-invariants of matrices. *CoRR*, abs/1801.02043, 2018.

[DM19] Harm Derksen and Visu Makam. An exponential lower bound for the degrees of invariants of cubic forms and tensor actions, 2019.

[DMR11a] Hang Dinh, Cristopher Moore, and Alexander Russell. McEliece and Niederreiter cryptosystems that resist quantum Fourier sampling attacks. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 761–779, 2011.

[DMR11b] Hang Dinh, Cristopher Moore, and Alexander Russell. Quantum Fourier sampling, code equivalence, and the quantum security of the McEliece and Sidelnikov cryptosystems. *CoRR*, abs/1111.4382, 2011.

[DMR15] Hang T. Dinh, Cristopher Moore, and Alexander Russell. Limitations of single coset states and quantum algorithms for code equivalence. *Quantum Information & Computation*, 15(3&4):260–294, 2015.

[DW00] Harm Derksen and Jerzy Weyman. Semi-invariants of quivers and saturation for littlewood-richardson coefficients. *Journal of the American Mathematical Society*, 13(3):467–479, 2000.

[FG19] Luca De Feo and Steven D. Galbraith. Seasign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 759–789. Springer, 2019.

[FGS19] Vyacheslav Futorny, Joshua A. Grochow, and Vladimir V. Sergeichuk. Wildness for tensors. *Linear Algebra and its Applications*, 566:212 – 244, 2019.

[Fis05] Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In *Advances in Cryptology – Crypto 2005*, pages 152–168. Springer, 2005.

[FP06] Jean-Charles Faugère and Ludovic Perret. Polynomial equivalence problems: Algorithmic and theoretical aspects. In *Advances in Cryptology - EUROCRYPT 2006*, pages 30–47, 2006.

[FR85] Katalin Friedl and Lajos Rónyai. Polynomial time solutions of some problems of computational algebra. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 153–162. ACM, 1985.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO 1986*, pages 186–194, 1986.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[GMS03] Willi Geiselmann, Willi Meier, and Rainer Steinwandt. An attack on the isomorphisms of polynomials problem with one secret. *Int. J. Inf. Sec.*, 2(1):59–64, 2003.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.

[Gro12] Joshua A. Grochow. Matrix isomorphism of matrix lie algebras. In *Proceedings of the 27th Conference on Computational Complexity, CCC 2012, Porto, Portugal, June 26-29, 2012*, pages 203–213, 2012.

[GV18] Steven D Galbraith and Frederik Vercauteren. Computational problems in supersingular elliptic curve isogenies. *Quantum Information Processing*, 17(10):265, 2018.

[Hås90] Johan Håstad. Tensor rank is NP-complete. *J. Algorithms*, 11(4):644–654, 1990.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HL13] Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *J. ACM*, 60(6):45:1–45:39, 2013.

[HLY12] Yun-Ju Huang, Feng-Hao Liu, and Bo-Yin Yang. Public-key cryptography from new multivariate quadratic assumptions. In *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, pages 190–205, 2012.

[HMR+10] Sean Hallgren, Cristopher Moore, Martin Rötteler, Alexander Russell, and Pranab Sen. Limitations of quantum coset states for graph isomorphism. *J. ACM*, 57(6):34:1–34:33, November 2010.

[HR14] Ishay Haviv and Oded Regev. On the lattice isomorphism problem. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 391–404, 2014.

[HS07] Rupert J. Hartung and Claus-Peter Schnorr. Public key identification based on the equivalence of quadratic forms. In *Mathematical Foundations of Computer Science 2007, 32nd International Symposium, MFCS 2007, Ceský Krumlov, Czech Republic, August 26-31, 2007, Proceedings*, pages 333–345, 2007.

[IKS10] Gábor Ivanyos, Marek Karpinski, and Nitin Saxena. Deterministic polynomial time algorithms for matrix completion problems. *SIAM J. Comput.*, 39(8):3736–3751, 2010.

[IQ19] G. Ivanyos and Y. Qiao. Algorithms based on *-algebras, and their applications to isomorphism of polynomials with one secret, group isomorphism, and polynomial identity testing. *SIAM Journal on Computing*, 48(3):926–963, 2019.

[IQS17] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Constructive non-commutative rank computation is in deterministic polynomial time. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 55:1–55:19, 2017.

[Kay11] Neeraj Kayal. Efficient algorithms for some special cases of the polynomial equivalence problem. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1409–1421, 2011.

[KLLNP16] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In *Advances in Cryptology – CRYPTO 2016*, pages 207–237. Springer, 2016.

[KLS18] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In *Advances in Cryptology – EUROCRYPT 2018*, pages 552–586. Springer, 2018.

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

[LQ17] Yinan Li and Youming Qiao. Linear algebraic analogues of the graph isomorphism problem and the Erdős-Rényi model. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 463–474, 2017.

[Luk82] Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.*, 25(1):42–65, 1982.

[LW54] Serge Lang and André Weil. Number of points of varieties in finite fields. *American Journal of Mathematics*, 76(4):819–827, 1954.

[LW12] Mark L. Lewis and James B. Wilson. Isomorphism in expanding families of indistinguishable groups. *Groups Complex. Cryptol.*, 4(1):73–110, 2012.

[McK80] Brendan D. McKay. Practical graph isomorphism. *Congr. Numer.*, pages 45–87, 1980.

[MFK94] David Mumford, John Fogarty, and Frances Kirwan. *Geometric invariant theory.* Springer-Verlag, 1994.

[MP14] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *J. Symb. Comput.*, 60:94–112, 2014.

[MPG13] Gilles Macario-Rat, Jérôme Plût, and Henri Gilbert. New insight into the isomorphism of polynomial problem IP1S and its use in cryptography. In *Advances in Cryptology - ASIACRYPT 2013*, pages 117–133, 2013.

[MRV07] Cristopher Moore, Alexander Russell, and Umesh Vazirani. A classical one-way function to confound quantum adversaries. *arXiv preprint quant-ph/0701115*, 2007.

[MT08] Ueli Maurer and Stefano Tessaro. Basing PRFs on constant-query weak PRFs: Minimizing assumptions for efficient symmetric cryptography. In *Advances in Cryptology – ASIACRYPT 2008*, pages 161–178. Springer, 2008.

[Mul17] Ketan D. Mulmuley. Geometric complexity theory V: Efficient algorithms for Noether normalization. *Journal of the American Mathematical Society*, 30(1):225–309, 2017.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of cryptology*, 4(2):151–158, 1991.

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudorandom functions. *J. ACM*, 51(2):231–262, 2004.

[O'B94] Eamonn A O'Brien. Isomorphism testing for p-groups. *Journal of Symbolic Computation*, 17(2):133–147, 1994.

[Pat96] Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In *Advances in Cryptology - EUROCRYPT 1996*, pages 33–48, 1996.

[Per05] Ludovic Perret. A fast cryptanalysis of the isomorphism of polynomials with one secret problem. In *Advances in Cryptology - EUROCRYPT 2005*, pages 354–370, 2005.

[PFM14] Jérôme Plût, Pierre-Alain Fouque, and Gilles Macario-Rat. Solving the "isomorphism of polynomials with two secrets" problem for all pairs of quadratic forms. *CoRR*, abs/1406.3163, 2014.

[PGC98] Jacques Patarin, Louis Goubin, and Nicolas Courtois. Improved algorithms for isomorphisms of polynomials. In *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, pages 184–200, 1998.

[PR97] Erez Petrank and Ron M. Roth. Is code equivalence easy to decide? *IEEE Trans. Information Theory*, 43(5):1602–1604, 1997.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of cryptology*, 13(3):361–396, 2000.

[Reg04] Oded Regev. Quantum computation and lattice problems. *SIAM J. Comput.*, 33(3):738–760, 2004.

[Sen00] Nicolas Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Trans. Information Theory*, 46(4):1193–1203, 2000.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 124–134, 1994.

[Sim78] Charles C Sims. Some group-theoretic algorithms. In *Topics in algebra*, pages 108–124. Springer, 1978.

[SS13] Nicolas Sendrier and Dimitris E. Simos. The hardness of code equivalence over and its application to code-based cryptography. In *Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013, Limoges, France, June 4-7, 2013. Proceedings*, pages 203–216, 2013.

[SY17] Fang Song and Aaram Yun. Quantum security of NMAC and related constructions. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, pages 283–309, 2017.

[Unr15] Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *Advances in Cryptology – EUROCRYPT 2015*, pages 755–784. Springer, 2015.

[Unr16] Dominique Unruh. Computationally binding quantum commitments. In *Advances in Cryptology – Eurocrypt 2016*, pages 497–527. Springer, 2016.

[Unr17] Dominique Unruh. Post-quantum security of Fiat-Shamir. In *Advances in Cryptology – Asiacrypt 2017*, pages 65–95. Springer, 2017.

[Wil09] James B. Wilson. Decomposing $p$-groups via Jordan algebras. *Journal of Algebra*, 322(8):2642–2679, 2009.

[WL68] Boris Weisfeiler and Andrei A. Lehman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2(9):12–16, 1968.

[Zha12] Mark Zhandry. How to construct quantum random functions. In *FOCS 2012*, pages 679–687, 2012.