

The Graded Monoidal Action for Cryptography

Jonathan Komada Eriksen^{1,2} and Emil August Hovd Olaisen³

¹ Nasjonal sikkerhetsmyndighet, Kolsås, Norway

² COSIC, KU Leuven, Belgium

³ Norwegian University of Science and Technology, Trondheim, Norway

Dedicated to the memory of H. M. Kong Harald V. Takk for alt.

Abstract. We give a new framework for constructing post-quantum protocols based on graded monoidal actions. An essential difference between the graded monoidal action framework and the cryptographic group action framework is that our higher-rank problems give rise to infinite structures, where elements do not admit inverses, while cryptographic group actions are finite by definition. Nevertheless, we show that the hard problems for cryptographic group actions reduces to the corresponding family of hard problems for graded monoidal actions. However, the main motivation is that for general graded monoidal actions, the relevant problems admit no known subexponential quantum attacks, while still having enough structure to formulate some protocols based on cryptographic group actions, such as the Diffie-Hellman key-exchange. Our framework can, to the extent we make clear in the paper, be instantiated with the module-action on oriented principally polarized abelian varieties (PPAV's), generalizing the class group action on oriented elliptic curves. Although the theory involving the module-action is quite involved, the formulation of a graded monoidal action is strikingly simple, making it a suitable abstraction for constructing protocols in, which can then be instantiated with the module-action.

1 Introduction

In 1976, Diffie and Hellman introduced the Diffie-Hellman key-exchange [14], which started the field of public-key cryptography. Despite being the first public-key protocol, it is still arguably the most important, with countless other cryptographic protocols being built on it. However, the original protocol worked in the cyclic group $(\mathbb{F}_p)^\times$. Due to the Index-Calculus attack, which gives a sub-exponential attack by exploiting the finite-field structure [26], one needs to use large sizes of the primes p to instantiate the scheme in a secure way, which lowers the performance significantly.

Luckily, the Diffie-Hellman protocol can be formulated just as well in any finite cyclic group. The advantage of this, is that one avoids the index-calculus attack, as it crucially relies on the finite-field structure. Thus, today the Diffie-Hellman protocol is rarely implemented in the whole group $(\mathbb{F}_p)^\times$. Instead, if one still wishes to in the group of units of a finite field, one can choose a low-order

element $g \in (\mathbb{F}_p)^\times$ (but still keeping p big enough to avoid Index-Calculus, and the order of g high enough to avoid generic group attacks), and work in the cyclic group $\langle g \rangle$. Even better, one can work over other algebraic groups as well, with the by far most common choice being to work in the cyclic group generated by a point $P \in E/\mathbb{F}_p$, for some elliptic curve $E/\mathbb{F}_p$ [24, 21]. Today, the overwhelming majority of Diffie-Hellman instantiations happens in these groups, for efficiency reasons [22].

However, even in generic cyclic groups, the Diffie-Hellman protocol is broken by Shor’s algorithm [34], a quantum polynomial-time algorithm. Thus, one needs to generalise further, if one wishes to avoid this attack. And precisely this is achieved by instead using a cryptographic group action, which is still sufficient to construct an analogue of the Diffie-Hellman protocol. This generalizes the cyclic group situation, as for any cyclic group G of order n , we have a regular action of $(\mathbb{Z}/n\mathbb{Z})^\times$ on the set of generators of G , and the commutativity of this action is essentially what gives the Diffie-Hellman key exchange. This was first noted by Couveignes [10] (and later, independently by Rostovtsev and Stolbunov [33]), with the introduction of the CRS cryptosystem.

Today, a practical instantiation of the CRS cryptosystem is made possible through optimizations introduced in CSIDH [8], with closer analogues of the original formulation of Hard Homogeneous Spaces having been made possible through (qt)-PEGASIS [11, 12], which were based on the techniques introduced in Clapote [28]. Although the underlying theory behind all these works is quite mathematically involved, the abstraction of Efficient Group Actions (EGAs) [1] makes it easy to formulate post-quantum analogues of Diffie-Hellman protocols, which can then be instantiated with CSIDH, or (qt)-PEGASIS.

However, even the formulation through cryptographic group actions is vulnerable to the quantum subexponential attack by Kuperberg [23]. This again leads to a need for large parameter-sizes for CSIDH and (qt)-PEGASIS, which lowers the performance on schemes based on such cryptographic group actions significantly.

1.1 Contributions.

In this work, we introduce the notion of an Efficient Graded Monoidal Action (EGMA), which, although not obvious, can be seen as a generalization of cryptographic group actions. The cryptographic graded monoidal action still has sufficient structure to instantiate a version of the Diffie-Hellman key-exchange, but (conjecturally), the higher-rank problems arising in this framework, do not have enough structure to run Kuperberg’s algorithm. As supporting evidence of this conjecture, a result by Robert implies that the general isogeny-problem (which forms the basis for many post-quantum protocols which have claimed no subexponential attacks, such as SQIsign [13]), reduces to $(1, 2)$ -graded vectorization (see Theorem 5.10), one of the natural higher-rank generalizations of vectorization in this framework.

The formulation of the graded monoidal action is based on the work by Robert [30], where he introduces the module action. In fact, our axioms can

simply be seen as a de-categorification of the module action framework introduced by Robert (see Theorem 5.1). Although the mathematics underlying the module action is highly involved, our formulation of a graded monoidal action is very simple, but is meant to capture the essential cryptographic properties of the module-action, similar to how cryptographic group actions captures the properties of the class-group action coming from complex multiplication theory underlying CRS. This formulation is meant for cryptographers to create cryptographic protocols, which can be instantiated with the module-action, without having to learn all the details of the module-action. The motivation for trying to formulate protocols in this framework rather than group actions, is the lack of a quantum subexponential algorithms, thus, one may choose much more aggressive parameters, which in turn typically leads to better performance. One example of this is the MIKE protocol, which has recently been implemented [2], and massively outperforms CSIDH, both in terms of computational complexity and public key sizes.

Related work. There is a line of research on semi-group actions for cryptography [32, 17], which at first seems slightly more general than a monoidal action, as a monoid is a semi-group with unit. However, our formulation of a graded monoidal action is surprisingly quite different than the one used in this line of work, with one essential difference being that the grading means that our monoids are infinite, while [32] only works with finite semi-groups, giving graded monoidal actions severely different properties. Despite big differences, there are some similarities as well: notably the sigma protocols we formulate (see Section 6.2) are quite similar to the ones in [32], and suffer from the same necessity of additional assumptions.

However, we note that, in practice, the biggest advantage is that we have a genuine instantiation of the graded monoidal action (which is conjecturally strictly harder than group actions), while this is to the best of our knowledge lacking for the semi-group action formulated in [32].

Concurrent work. This paper comes out at the same time as the paper by Basso, Dartois, Duparc, Eriksen, Kunzweiler, Meyer, Pope, Reijnders, Robert, Rueger, and Schaeffler [2], which works out the necessary theory to make the original MIKE protocol by Robert [30] practical, and provides an implementation. However, this recent work is quite detailed, and may be hard to digest for non-specialists. These remarks motivates this work, as the results from the MIKE paper [2] look highly promising for constructing other post-quantum protocols, while the technical nature of the paper might be an obstacle for non-specialists. Another paper that comes out at the same time as this work, is the paper by Eriksen, Invernizzi, Sferlazza, Tasbihgou, and Vercauteren [16], which, among other things, analyses the instantiation of an EGMA based on the full module action of Robert, and shows that one can always recover the Steinitz class of the rank-2 module, thus further motivating the *skeleton* (see Definition 3.6), which is for instance used in MIKE.

Structure of the paper. In Section 2, we recall some necessary preliminaries. In Section 3 we introduce the necessary algebraic properties of our graded monoidal actions, before defining EGMA, and discussing relevant cryptographic problems in Section 4. In Section 5, we discuss to what extent the module-action satisfies the properties of an EGMA, before we finally give some examples of some cryptographic constructions based on EGMA in Section 6.

Acknowledgements. First and foremost, thanks to the whole MIKE team for plenty of great discussions on the module action. Several results, e.g. Lemma 5.8, were formulated as a direct result of discussions with the MIKE team. Further, thanks to everyone at LID 2025 who took an interest in the topic for brainstorming. Thanks to everyone who worked on the extraction problem at the Isogeny Club - Brainstorm Days 2026 as well, which provided useful discussions.

This work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement ISOCRYPT – No. 101020788), by the Research Council KU Leuven grant C14/24/099 and by CyberSecurity Research Flanders with reference number VOEWICS02.

2 Preliminaries

Notation $\mathbb{N}$ will refer to the set of non-zero, positive integers. Unless otherwise stated, we will use $\cdot$ to denote a group multiplication, and $\otimes$ to denote monoid multiplication. In either case $\star$ is used as the operation of a group or monoid on a set.

All our cryptographic objects are implicitly families indexed by a security parameter λ . An algorithm (or adversary) is called probabilistic polynomial time (PPT), or simply *efficient* if its running time is polynomial in λ . Ranks and dimensions appearing in a protocol are regarded as fixed. For a finite set Y , the notation $y \leftarrow Y$ denotes uniform sampling, whereas $y \leftarrow \chi$ denotes sampling from a specified distribution χ on Y . Computational indistinguishability is always with respect to PPT adversaries and the implicit security parameter.

2.1 Monoidal actions

Recall that a monoid $(M, \otimes)$ is a set M with a specified element $e \in M$, together with a binary operation $\otimes : M \times M \rightarrow M$ such that the operation is associative, i.e. $(a_1 \otimes a_2) \otimes a_3 = a_1 \otimes (a_2 \otimes a_3)$, and such that e acts as the identity, i.e. $e \otimes a_1 = a_1 \otimes e = a_1$ for all $a_1, a_2, a_3 \in M$. A monoid is said to be commutative if the operation additionally satisfies $a_1 \otimes a_2 = a_2 \otimes a_1$ for all $a_1, a_2 \in M$.

A monoidal action $(M, X, \star)$ is a monoid M together with a set X and a binary operation $\star : M \times X \rightarrow X$ which respects the monoid-operation, i.e. $a_1 \star (a_2 \star x) = (a_1 \otimes a_2) \star x$ and $e \star x = x$ for all $a_1, a_2 \in M$ and $x \in X$.

A group is a monoid, where every element has an inverse, i.e. for all $a \in M$, there exists an a^{-1} such that $a \otimes a^{-1} = a^{-1} \otimes a = e$. A group action is then simply a monoidal action, where M is a group.

2.2 Cryptographic group actions

Early formulations of group actions with additional properties making them useful for cryptography can be found in [6] under the name *one-way group actions*, and in [10] as *hard-homogeneous spaces*. These formulations are not equivalent, as noted in [1]. They also further formalize some of the notions from [6, 10] into *effective group actions*.

We now recall some standard properties of group actions.

Definition 2.1. Let $(G, X, \star)$ be a group action. We say that the action is:

- Transitive:** if, for every $x_1, x_2 \in X$, there exists $g \in G$ such that $g \star x_1 = x_2$;
- Faithful:** if, for every $g \in G$, the condition $g \star x = x$ for every $x \in X$ implies that $g = e$;
- Free:** if, for every $g \in G$ and $x \in X$, the equality $g \star x = x$ implies that $g = e$;
- Regular:** if it is both free and transitive.

Every free group action is faithful. Moreover, for every $x \in X$, the map

$$\pi_x : G \longrightarrow X, \quad g \longmapsto g \star x,$$

is a bijection if the group action is regular.

To use a group action in a cryptographic scheme, one needs efficient algorithms for the relevant operations, together with hard computational problems. With this in mind, we recall the definition of an effective group action from [1].

Definition 2.2. A group action $(G, X, \star)$ is an *effective group action (EGA)* if the following properties are satisfied.

Algorithms for G : The group G is finite, and there exist PPT algorithms for:

- membership testing in G ;
- equality testing in G ;
- sampling from a distribution χ_G on G which is statistically close to uniform;
- computing $g \cdot h \in G$, for every $g, h \in G$;
- computing $g^{-1} \in G$, for every $g \in G$.

Algorithms for X : The set X is finite, and there exist PPT algorithms for:

- membership testing in X ;
- computing a canonical representation of any $x \in X$.

Origin: There exists a distinguished element $x_0 \in X$, called the *origin*, with a known representation.

Algorithm for the action: There exists a PPT algorithm which, given $g \in G$ and $x \in X$, computes $g \star x \in X$.

It is worth noting that CSIDH [8] is not an effective group action. It is, however, a *Restricted Effective Group Action (REGA)*, a notion also introduced in [1]. However, (qt)-PEGASIS [11, 12] has shown how to construct an EGA from the class-group action underlying CSIDH. We choose not to elaborate on the differences between the two, as the module action we introduce in Section 5 is not a restricted action in the sense of a REGA.

2.3 The module action

In this section, we recall a few preliminary results needed for Section 5, where we show that the module actions satisfies the properties we are interested in from a graded monoidal action, before we prove a few more properties, which generally will depend on how the graded monoidal action is instantiated. These preliminaries are not needed outside of Section 5, thus readers who are less interested in how to instantiate the graded monoidal action is welcomed to skip this part.

For now, let R be a Dedekind-domain, with field of fractions K . The following theorem characterizes finitely generated projective modules over R

Theorem 2.3. *Let P be a finitely generated projective R -module of rank n . Then*

$$P \cong R^{n-1} \oplus \mathfrak{a},$$

where $\mathfrak{a}$ uniquely determines an element $[\mathfrak{a}] \in \mathcal{C}(R)$.

Proof. See e.g. [9, Theorem 1.2.19]. □

Thus, a projective R -module P is uniquely determined by its rank and the ideal class $[\mathfrak{a}]$ in the theorem above. This ideal class is called the *Steinitz class* of $\mathfrak{a}$, and is denoted $\text{St}(P)$. Most practical computations for projective R -modules follow from the existence of pseudobases.

Theorem 2.4. *Let P be a finitely generated projective R -module of rank n . There exists elements $e_1, \dots, e_n \in P$, and fractional ideals $\mathfrak{a}_1, \dots, \mathfrak{a}_n \subset K$ such that*

$$P \simeq \mathfrak{a}_1 e_1 \oplus \dots \oplus \mathfrak{a}_n e_n,$$

and $[\mathfrak{a}_1 \dots \mathfrak{a}_n] = \text{St}(P)$.

Proof. See e.g. [9, Corollary 1.2.25]. □

We call the tuple $((e_1, \dots, e_n), (\mathfrak{a}_1, \dots, \mathfrak{a}_n))$ a *pseudobasis* of P .

From now, let $K = \mathbb{Q}(\sqrt{-p})$ for some positive integer p , and let $R = \mathcal{O}_K$, the ring of integers. Then R is Dedekind, and comes with an involution corresponding to complex conjugation.

A *Hermitian form* H on P is a map $H : P \times P \rightarrow R$, which is R -linear on the first component, and R -antilinear on the second, satisfying $H(x, y) = \overline{H(y, x)}$. We will be concerned with *unimodular, positive-definite* Hermitian forms. Unimodularity means that

$$\begin{aligned} P &\rightarrow \text{Hom}_{\overline{R}}(P, R) \\ x &\rightarrow H(x, -) \end{aligned}$$

is an isomorphism, while positive-definite means that $H(x, x) > 0$ for all $0 \neq x \in P$ (note that the Hermitian condition implies that $H(x, x) = \overline{H(x, x)}$ implying that $H(x, x) \in \mathbb{Z}$).

For practical computations, we pick a pseudo-basis of P , which induces a gram matrix G for H , i.e

$$G = (g_{ij}), \quad g_{ij} = H(e_i, e_j).$$

However, the pseudobasis is important for the context of the Gram-matrix, thus we can define the *Gram-data* to be the pair $(G, (\mathfrak{a}_1, \dots, \mathfrak{a}_n))$. Unimodularity and positive-definiteness can then be checked by the following theorem. We state the theorem for general rings of integers in CM-orders, though we will only need the quadratic case (in the general case, the last statement must then be extended to checking the condition for all complex embeddings).

Theorem 2.5. *Let R be the ring of integers in a CM field, and let P be a finitely generated projective R -module of rank n , with chosen pseudo-basis $((e_1, \dots, e_n), (\mathfrak{a}_1, \dots, \mathfrak{a}_n))$. Let H be a sesquilinear form on P , and write the induced Gram-data as $(G, (\mathfrak{a}_1, \dots, \mathfrak{a}_n))$. Then:*

- H is hermitian $\Leftrightarrow \overline{G} = G^T$.
- H is unimodular $\Leftrightarrow \prod_i \mathfrak{a}_i \det(G) \prod_i \overline{\mathfrak{a}_i} = R$.
- In the case that R is quadratic, H is positive-definite $\Leftrightarrow G$ is positive definite as a matrix over $\mathbb{C}$ (under the unique pair of complex embeddings $\text{Frac}(R) \hookrightarrow \mathbb{C}$).

Proof. Follows from the material in [20, Section 2.2-2.4]. □

Note that the condition for unimodularity is easier stated as

$$\det(G) \mathfrak{n}(\mathfrak{a}_1) \dots \mathfrak{n}(\mathfrak{a}_n) \in R^\times.$$

Finally, we recall the anti-equivalence of categories underlying the module-action of Robert. For a background on abelian varieties, see the book by Milne [25].

Let $E_0/\mathbb{F}_p$ be an elliptic curve with a (necessarily primitive) R -orientation, i.e. $R \hookrightarrow \text{End}(E_0)$. Let $\mathcal{A}_{R, E_0}$ denote the category of R -oriented abelian varieties, that are R -isogenous to E_0^n , for some n .

Theorem 2.6. *There exists an anti-equivalence of categories between the category of finitely generated projective R -modules and $\mathcal{A}_{R, E_0}$. On objects, the anti-equivalence sends $M \rightarrow M \cdot E_0$, and satisfies $(M \otimes N) \cdot E_0 = M \cdot (N \cdot E_0)$.*

Proof. See [30, Theorem 4.9] for the anti-equivalence. For the second part, see [30, Example 3.8]. □

Under the anti-equivalence, a unimodular positive-definite Hermitian form H on M corresponds to a principal polarization $\lambda : A \rightarrow A^\vee$, see [30, Theorem 4.11].

3 Graded Monoidal Actions and their Properties

In this section, we discuss graded monoidal actions and several properties we are interested in, in preparation for the next section, where we formally define our framework, the *Effective Graded Monoidal Action* (EGMA).

We start with the following definition.

Definition 3.1. *A (multiplicatively) graded monoid is a monoid $(M, \otimes)$ whose underlying set can be written as a disjoint union $M = \bigsqcup_{i \in \mathbb{N}} M_i$ (as a set) such that for each $r, s \in \mathbb{N}$ the monoid-operation restricts to a map:*

$$\otimes : M_r \times M_s \rightarrow M_{rs}.$$

Similarly, a (multiplicatively) graded monoidal action $(M, X, \star)$ is a monoidal action by a (multiplicatively) graded monoid M on a set $X = \bigsqcup_{i \in \mathbb{N}} X_i$, such that for each $r, s \in \mathbb{N}$ the action restricts to a map:

$$\star_{r,s} : M_r \times X_s \rightarrow X_{rs}.$$

Remark 3.2. For some readers, it might be helpful to analyse the definition of a graded monoid as an analogue of the definition of a graded ring in commutative algebra. Indeed, the set of non-zero homogeneous elements of a graded ring without zero-divisors⁴ forms a graded monoid. To see this, note that if

$$R = \bigoplus_{i \in \mathbb{N}} R_i$$

is a multiplicatively graded integral domain with $1_R \in R_1$, then the multiplicative structure satisfies:

$$R_r \cdot R_s \subseteq R_{rs},$$

For the more common additive grading (which we do not use, but where we would need $1_R \in R_0$), indexed by $\mathbb{N} \cup \{0\}$, the corresponding condition is

$$R_r \cdot R_s \subseteq R_{r+s}$$

Thus, the non-zero homogeneous elements of a graded integral domain form a graded monoid with respect to multiplication.

Similarly, if:

$$N = \bigoplus_{i \in \mathbb{N}} N_i$$

is a torsion-free graded module over a multiplicatively graded integral domain R , then the module action satisfies

$$R_r \cdot N_s \subseteq N_{rs}$$

⁴ This assumption is important, as otherwise, we might have $r_1 r_2 = 0$ even for homogeneous elements.

(or, additively graded, $R_r \cdot N_s \subseteq N_{r+s}$). Hence, the non-zero homogeneous elements of R act on the non-zero homogeneous elements of N , giving a graded monoidal action equipped with additional additive structure compatible with the action.

For the rest of this paper, a monoid M (resp. a set X) will always decompose as a disjoint union $M = \bigsqcup_{i \in \mathbb{N}} M_i$ (resp. $X = \bigsqcup_{i \in \mathbb{N}} X_i$). All gradings from this point forward will be multiplicative. We will refer to the grading of an element in M as its rank, i.e. we say that a has rank d if $a \in M_d$, and similarly, we refer to the grading of an element in X as its dimension.

Our goal is to generalise finite, free, and transitive group actions. As noted in the preliminaries, one useful consequence of this is that for any free and transitive group action and any fixed $x_0 \in X$, the map $G \rightarrow X$ given by $a \mapsto a \star x_0$ is a bijection.

The natural generalisation of being free, namely that for all $x \in X$, and all $a, b \in M$ satisfying $a \star x = b \star x$, we have $a = b$, is too strong. To see this, note that it is not even clear that this holds for M alone. If for all $a, b, c \in M$, we have that $a \otimes c = b \otimes c$ implies $a = b$, M is said to be *cancellable*, which is a very strong property on M we do not want to impose.

Even worse, neither finiteness nor transitivity generalises directly at all. To see this, observe that M cannot be finite as soon as $M \setminus M_1$ is non-empty, since for any $a \in M_d$, with $d > 1$, the elements

$$a, a \otimes a, a \otimes a \otimes a, \dots$$

are all distinct, since they have strictly increasing rank. Similarly, transitivity is not possible for graded monoidal actions: if $x_1 \in X_{d_1}$ and $x_2 \in X_{d_2}$ with $d_2 < d_1$, then there is no $a \in M$ such that $a \star x_1 = x_2$.

Thus, we give the following three definitions, which we argue are the “right” generalizations of free, finite and transitive in our setting.

Definition 3.3. *Let M be a graded monoid, and $(M, X, \star)$ a graded monoidal action.*

- *M is called rankwise finite if M_d is finite for all $d \in \mathbb{N}$.*
- *$(M, X, \star)$ is called base-free if for all $x \in X_1$, and $a, b \in M$, $a \star x = b \star x$ implies that $a = b$.*
- *$(M, X, \star)$ is said to be base-transitive, if for all $x \in X_1$ and $y \in X$, there exists an $a \in M$ such that $y = a \star x$.*

A first trivial observation is that this recovers the bijections in each rank and corresponding degree.

Lemma 3.4. *Let $(M, X, \star)$ be a rankwise finite, base-free and base-transitive graded monoidal action. Then for any $x_0 \in X_1$, and any $d \in \mathbb{N}$, we have bijections of finite sets*

$$\begin{aligned} M_d &\rightarrow X_d \\ a &\rightarrow a \star x_0 \end{aligned}$$

Proof. The map is injective by base-freeness, and surjective by base-transitivity. Finally, M_d is finite by rankwise finiteness, and hence so is X_d . $\square$

We now give a second observation, relating these graded monoidal actions to free and transitive group actions.

We start with the following:

Proposition 3.5. *If $(M, X, \star)$ is a commutative, base-free and base-transitive graded monoidal action, then:*

- M_1 is an abelian group.
- M_1 acts (not necessarily freely) on X .
- The restricted action:

$$\star_{1,1} : M_1 \times X_1 \rightarrow X_1$$

is a free and transitive commutative group action.

Proof. Fix an element $x \in X_1$, and let $a \in M_1$. By base-transitivity, applied with base point $a \star x \in X_1$ and target x , there exists a unique $b \in M_1$ such that $x = b \star (a \star x) = (b \otimes a) \star x$, where uniqueness follows from base-freeness. Base-freeness also implies that $b \otimes a = 1_M$, so every element of M_1 is invertible. Thus M_1 is a group, which is abelian since M is commutative, and the two subsequent statements follow immediately. $\square$

The M_1 -structure means that we can group X into M_1 -orbits. We can visualize this as a disjoint union of Schreier graphs (see Figure 1). As the proposition specifies, these orbits need not all look the same, since the action of M_1 is only guaranteed to be free in dimension 1 (i.e., only the orbit X_1 is guaranteed to be a Cayley graph). However, we will later give an easy criterion that is sufficient to get a free action of M_1 on X_d in the specific instantiation we have in mind (see Lemma 5.8).

By Proposition 3.5, M_1 is a group that acts on X . Define an equivalence relation on X by $x \sim_{M_1} y$ whenever there exists an $a \in M_1$ such that $x = a \star y$. The induced action⁵ by M/M_1 on $X/\sim_{M_1}$, which we denote by $\bar{\star}$, is then well defined. Indeed, let $a \in M$, let $a_1, a_2 \in M_1$, and let $x \in X$. Replacing a by $a \otimes a_1$ and x by $a_2 \star x$ does not change the resulting equivalence class, since

$$[a \otimes a_1] \bar{\star} [a_2 \star x] = [(a \otimes a_1) \star (a_2 \star x)] = [(a_1 \otimes a_2) \star (a \star x)] = [a \star x] = [a] \bar{\star} [x].$$

Thus, we give the following definition:

Definition 3.6. *Given a rankwise finite, base-free and base-transitive graded monoidal action $(M, X, \star)$, we call the graded monoidal action $(M/M_1, X/\sim_{M_1}, \bar{\star})$ the skeleton of $(M, X, \star)$.*

Quotienting by M_1 removes the degree-1 group-action component. The skeleton therefore isolates the part of the action to which attacks exploiting that group structure do not directly apply. We can visualize it as in Figure 2.

⁵ M/M_1 is simply the standard quotient of monoids, but since M_1 is a group, it is conceptually particularly simple: the elements of M/M_1 are equivalence classes of M under the relation $a \sim b$ if $a = b \otimes c$ for some $c \in M_1$.

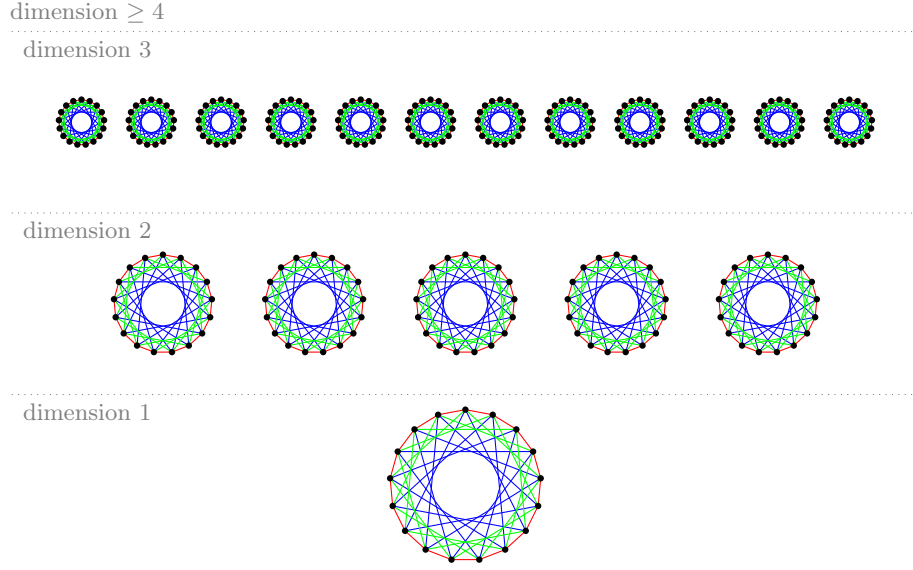


Fig. 1. A rankwise finite, base-free and base-transitive graded monoidal action (up to dimension 3), where the colored edges correspond to action by three different rank-1 elements.

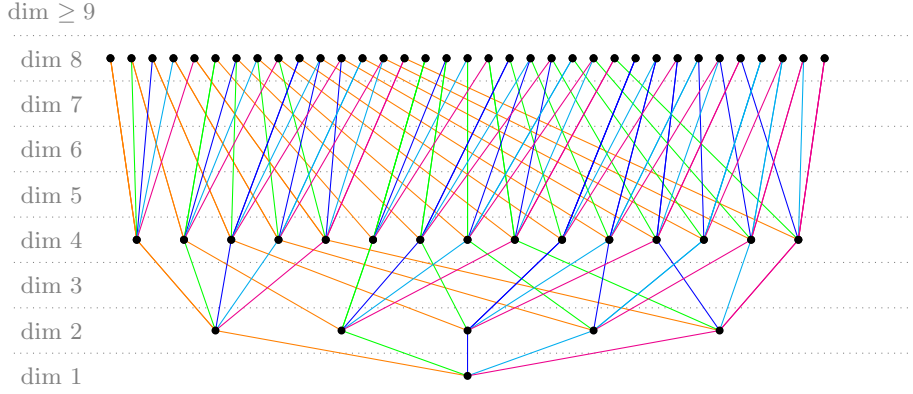


Fig. 2. The skeleton of a rankwise finite, base-free and base-transitive graded monoidal action (up to dimension 8). Only the part generated by 5 rank-2 elements of M is shown. Colored edges (which are directed upwards) correspond to the action by these elements. Notice that each path passing over the same colored edges ends in the same node. This is the commutativity of M .

Remark 3.7. In Figure 2, it seems like for instance the elements of dim 4 can be reached in a unique way up to ordering, i.e. for $a_1, a_2, b_1, b_2 \in M_2$, we have that $a_1 \otimes a_2 = b_1 \otimes b_2$ implies that $\{a_1, a_2\} = \{b_1, b_2\}$. We warn again that this need not hold in general.

We now describe a simple way of turning a group action into a monoidal action which will prove useful later for our security reductions.

Definition 3.8. *Given a free and transitive group action $(G, X, \star)$, we can construct the graded monoid $\tilde{G} := \mathbb{N} \times G$, with operation*

$$(r, g) \otimes (s, h) = (rs, g \cdot h),$$

identity $(1, e_G)$, and grading $\tilde{G} = \bigsqcup_{i \in \mathbb{N}} (i, G)$. In this way, $\tilde{G}$ acts base-freely (or in this case, even freely) and base-transitively on the graded set $\tilde{X} := \mathbb{N} \times X = \bigsqcup_{i \in \mathbb{N}} (i, X)$ by the action

$$\begin{aligned} \tilde{\star} : \tilde{G} \times \tilde{X} &\rightarrow \tilde{X} \\ ((r, g), (s, x)) &\rightarrow (rs, g \star x). \end{aligned}$$

We call $(\tilde{G}, \tilde{X}, \tilde{\star})$ the associated graded monoidal action of $(G, X, \star)$.

Given a free and transitive group action, the associated graded monoidal action simply has a skeleton that consists of a single element in each dimension.

4 Effective Graded Monoidal Actions

In this section, we define effective graded monoidal actions. The definition closely follows that of an effective group action in [1] (compare the below to Definition 2.2), with the algorithms indexed by the ranks in which they are required.

Definition 4.1. *Let $N \subseteq \mathbb{N}$ be a subset, with $1 \in N$. A base-free and base-transitive graded monoidal action $(M, X, \star)$ is an N -effective graded monoidal action (N -EGMA) if the following properties are satisfied.*

Algorithms for M : *The monoid M is commutative, rankwise finite, and there exist PPT algorithms for:*

- membership testing in M_d , for every $d \in N$;
- equality testing in M_d , for every $d \in N$;
- sampling from a distribution χ_d on M_d which is statistically close to uniform, for every $d \in N$;
- computing $a_1 \otimes a_2 \in M_{rs}$, for every $a_1 \in M_r$ and $a_2 \in M_s$ with $r, s, rs \in N$;
- computing $a^{-1} \in M_1$, for every $a \in M_1$.

Algorithms for X : *Each set X_d is finite, and there exist PPT algorithms for:*

- membership testing in X_d , for every $d \in N$;

– computing a canonical representation of any X_d , for every $d \in N$.

Origin: *There exists a distinguished element $x_0 \in X_1$, called the origin, with a known representation.*

Algorithm for the action: *There exists a PPT algorithm which, given $a \in M_r$ and $x \in X_s$ with $r, s, rs \in N$, computes $a \star x \in X_{rs}$.*

When N is understood from context, or $N = \mathbb{N}$, we may simply call the action an effective graded monoidal action (EGMA).

As in the definition of an effective group action, the sampling distributions are part of the description of the action. The condition $r, s, rs \in N$ means that an N -EGMA only includes operations whose inputs and outputs have degrees in N .

Most protocols only require N -effectiveness for a very small set $N \subseteq \mathbb{N}$, and in informal discussion we will often drop N from the terminology. The algorithms are not required to run in time polynomial in the rank or dimension. Thus, we may only be able to apply elements consecutively a few times before the dimension becomes too large for practical computation.

Remark 4.2. Let $\overline{M} = M/M_1$ and $\overline{X} = X/\sim_{M_1}$. The skeleton $(\overline{M}, \overline{X}, \overline{\star})$ of a rankwise finite, base-free and base-transitive graded monoidal action has the same properties and a trivial degree-1 monoid. However, even if $(M, X, \star)$ is an N -EGMA, showing that $(\overline{M}, \overline{X}, \overline{\star})$ is an N -EGMA requires argument: we must still be able to compute the induced equality (in both $\overline{M}$ and $\overline{X}$), and sampling in $\overline{M}$. However, if this is satisfied, we can apply the EGMA formalism directly to the skeleton. This may be preferable when a protocol only depends on M_1 -orbits. In this case, working with the skeleton removes the degree-1 group-action component and can give smaller representations and faster algorithms. Looking ahead (Theorem 5.6), it may even give new algorithms that were not already known.

4.1 Security assumptions

In this section, we define natural problems related to an EGMA. We stress that these problems may serve different purposes in different protocols. We will later see that the security proof of certain EGMA-protocols may rely both on the assumption that these problems are hard, or that there are efficient algorithms for them. These assumptions must then be analyzed for the specific instantiations of an EGMA.

Just as vectorization, parallelization, and group-action DDH generalize the discrete logarithm, computational Diffie–Hellman, and decisional Diffie–Hellman problems to the group-action setting, we can define natural analogues in the graded monoidal setting. Here they become families of problems indexed by compatible degrees.

Unlike for discrete logarithms and vectorization in a free group action, the solution to (r, s) -graded vectorization need not be unique when $r > 1$. Base-freeness guarantees uniqueness only when the input lies in X_1 .

Problem 4.3 ((r, s)-Graded Vectorization). Given $x \in X_r$ and $y \in X_{rs}$, with the guarantee that $y = a \star x$ for some $a \in M_s$, compute an $a' \in M_s$ satisfying $y = a' \star x$.

Problem 4.4 ((r, s, t)-Graded Parallelization). Given $(x_1, x_2, x_3) \in X_r \times X_{rs} \times X_{rt}$, with the guarantee that $x_2 = a_2 \star x_1$ and $x_3 = a_3 \star x_1$ for some $a_2 \in M_s$ and $a_3 \in M_t$, compute $(a_2 \otimes a_3) \star x_1 \in X_{rst}$.

Problem 4.5 ((r, s, t)-Graded DDH). Let $a_2, a'_2 \leftarrow \chi_s$ and $a_3, a'_3 \leftarrow \chi_t$ be independent. Given $x_1 \in X_r$, distinguish samples drawn from the distributions

$$(x_1, a_2 \star x_1, a_3 \star x_1, (a_2 \otimes a_3) \star x_1) \quad \text{and} \quad (x_1, a_2 \star x_1, a_3 \star x_1, (a'_2 \otimes a'_3) \star x_1).$$

By Proposition 3.5, $(r, 1)$ -graded vectorization, $(r, 1, 1)$ -graded parallelization, and $(r, 1, 1)$ -graded DDH are the ordinary vectorization, parallelization, and DDH problems, respectively, for the action of the group M_1 on an M_1 -orbit in X_r . Thus, a protocol intended to avoid attacks exploiting this group structure must involve secret monoid elements of rank at least 2. Already for $r = 1$ and $s = 2$, we can hope to obtain a significantly harder problem. For instance, a result of Robert [30], shows that the supersingular endomorphism-ring problem reduces to $(1, 2)$ -graded vectorization (see Theorem 5.10).

In the other direction, the associated graded monoidal action embeds the usual group-action problems.

Proposition 4.6. *For all $r, s, t \in \mathbb{N}$, vectorization reduces to (r, s) -graded vectorization, parallelization reduces to (r, s, t) -graded parallelization, and group-action DDH reduces to (r, s, t) -graded DDH.*

Proof. Let $(G, Y, \star)$ be the EGA where vectorization, parallelization, or DDH is instantiated, and consider the associated monoidal action $(\tilde{G}, \tilde{Y}, \tilde{\star})$ (see Definition 3.8). The fact that $(G, Y, \star)$ is an EGA immediately implies that $(\tilde{G}, \tilde{Y}, \tilde{\star})$ is an EGMA.

Given a vectorization instance $y_1, y_2 \in Y$, form the associated graded instance $(r, y_1), (rs, y_2)$. If $y_2 = g \star y_1$, its unique solution is $(s, g) \in \tilde{G}_s$, from which we recover g . Similarly, from a parallelization instance $y, y_1, y_2 \in Y$, form the graded instance $(r, y), (rs, y_1), (rt, y_2)$. The graded output has the form (rst, z) , where z is the solution to the original parallelization instance. Finally, the same degree tagging maps a group-action DDH instance (y, y_1, y_2, y_3) to $((r, y), (rs, y_1), (rt, y_2), (rst, y_3))$ and preserves both distributions. $\square$

Remark 4.7. We emphasize that the above proof is about *generic* algorithms for EGMA. Of course, it can happen for a specific instantiation of an EGMA that e.g. $(1, 1)$ -graded vectorization is hard, while $(1, d)$ -graded vectorization is easy for all $1 < d \in \mathbb{N}$. To see this, take the example where $(M_1, X_1, \star)$ is any EGA, while $M_d = \{\ast_d\} = X_d$, consists of a single element. Thus, it is not the case that for a *fixed* EGMA $(M, X, \star)$, the above reductions hold. For a fixed instance, differently graded problems need to be analyzed individually.

We now discuss two additional problems which are trivially settled for an EGA, but which will prove highly interesting for an EGMA. Since solutions to these problems need not be unique, we state two versions of each problems. We will later see that when efficient algorithms for these problems are necessary in security proofs, the full version will typically be the most interesting.

Problem 4.8 ((r, s)-Inversion). Given $a \in M_s$ and $y \in X_{rs}$ find *any* $x' \in X_r$ satisfying $y = a \star x'$, or decide that none exists.

Problem 4.9 (Full (r, s)-Inversion). Given $a \in M_s$ and $y \in X_{rs}$ find *all* $x' \in X_r$ satisfying $y = a \star x'$, or decide that none exists.

Note that (full) (r, s)-inversion is trivial for $s = 1$, since then a is invertible. For $s \geq 2$, there are no generic algorithms to solve this problem as far as we are aware. We can also consider inversion for the monoid M acting on itself by multiplication. We give this problem a special name, since its complexity will be important when constructing cryptographic schemes (see Section 6.2).

Problem 4.10 ((r, s)-extraction). Given $a \in M_s$ and $c \in M_{rs}$, find *any* $b' \in M_r$ satisfying $a \otimes b' = c$, or decide that none exists.

Problem 4.11 (full (r, s)-extraction). Given $a \in M_s$ and $c \in M_{rs}$, find *all* $b' \in M_r$ satisfying $a \otimes b' = c$, or decide that none exists.

Notice the asymmetry of these problems. In light of Proposition 3.5, (full) ($r, 1$)-extraction admits a polynomial-time algorithm for any $r \in \mathbb{N}$, using rank-1 inversion. To the best of our knowledge, there is no such algorithm for ($1, s$)-extraction in a general EGMA. We will see later that ($1, 2$)-extraction is efficiently solvable in one of the specific instantiations we consider (see Proposition 5.9).

Finally, we define a hiding problem which will very typically be a necessary assumption for security proofs. In the usual group-action setting, multiplying a uniformly random group element by a fixed element preserves the uniform distribution. For a graded monoid, this is no longer the case, and we must formulate the problem relative to the sampling distribution of the relevant EGMA. The public action value must also be included in the instance, since it is the information about the secret element available to an adversary.

Problem 4.12 ((r, s)-Hiding). Let $a, a' \leftarrow \chi_r$ and $b \leftarrow \chi_s$ be independent. Distinguish samples drawn from the distributions

$$(a \star x_0, a \otimes b) \quad \text{and} \quad (a \star x_0, a' \otimes b).$$

If $r = 1$ and χ_s is invariant under multiplication by M_1 (as is the uniform distribution on the finite set M_s), then the two distributions are identical, again using that a and a' are invertible. Thus, under this condition, the ($1, s$)-hiding problem is information-theoretically hard. If χ_s is only statistically close to an invariant distribution, the two distributions are correspondingly statistically close.

5 Relation to the Hermitian Module Action on Oriented PPAVs.

In this section we discuss an instantiation of the EGMA-framework, and we identify open problems to get a full EGMA.

Some of the material in this section is either already discussed in, or follows quickly from, the material in Robert's preprint on the module action [30]. Our purpose here is simply to link it to the graded module action we have discussed in this paper, and provide some results on the problems defined in Section 4.

We also note that this section requires preliminaries far beyond what is needed for the abstract graded monoidal action framework. Thus, readers who are primarily interested in the abstract framework, and less interested in how to instantiate it, are welcome to skip the current section (although Section 5.3 might still be of interest).

5.1 Hermitian modules acting on oriented principally polarized abelian varieties

Let R be the ring of integers in a CM field. As for any commutative ring, the set of isomorphism classes of finitely generated projective R -modules forms a graded monoid under the tensor product, where the grading is determined by the rank of the module P . Denote this monoid by $\mathcal{M}$. It is a classical result that all projective R -modules of rank 1 are invertible (this uses the maximality of R), and that the rank-1 part of $\mathcal{M}$ is in fact isomorphic to the class group $\mathcal{C}(R)$, see for instance the book by Weibel [35, Section 1.3].

For ease of exposition, let R now be the ring of integers in an imaginary quadratic number field $K = \mathbb{Q}(\sqrt{-p})$ for some prime p . Consider now an elliptic curve $E_0/\mathbb{F}_p$, together with a primitive R -orientation $R \hookrightarrow \text{End}(E_0)$, where $\sqrt{-p}$ is sent to the Frobenius endomorphism of E_0 . Denote by $\mathcal{X}$ the set of R -oriented isomorphism classes of abelian varieties A that are R -isogenous to E_0^n , for some n .

Theorem 5.1. *With the notation as above, there exists a well-defined binary operation*

$$\star : \mathcal{M} \times \mathcal{X} \rightarrow \mathcal{X}$$

making $(\mathcal{M}, \mathcal{X}, \star)$ into a base-free and base-transitive graded monoidal action. Further, $\mathcal{M}$ (and hence $\mathcal{X}$) is rankwise finite.

Further, let $\mathcal{M}'$ denote the set of positive-definite unimodular Hermitian R -modules up to isometry, and let $\mathcal{X}'$ denote the set of R -oriented isomorphism classes of PPAVs (A, λ_A, ι) that are R -isogenous (as principally polarized varieties) to E_0^n , for some n . Then, the action above induces a corresponding, efficiently computable action

$$\star' : \mathcal{M}' \times \mathcal{X}' \rightarrow \mathcal{X}'$$

with the same properties as above.

Proof. This all follows directly from Theorem 2.6, the aforementioned correspondence between Hermitian forms and polarizations [30, Theorem 4.11], and the work by Page, Robert, and Soumier [29]. To see this, note that the monoids and the sets in the statement are simply the isomorphism classes of the respective categories, and the fact that we have $(M \otimes N) \cdot E_0 = M \cdot (N \cdot E_0)$ (a symmetric monoidal action) induces a graded monoidal action. Further, note that rankwise-finite is implied by Steinitz theorem (see Theorem 2.3) for $\mathcal{M}$, and can be extended to $\mathcal{M}'$ by [20, Theorem 2.4.5], base-freeness is implied by the functor being fully faithful, while base-transitivity is implied by the functor being essentially surjective.

The fact that the second action is efficiently computable for any (fixed) rank and dimension was settled by Page, Robert, and Soumier [29, Theorem 6.5]. $\square$

Remark 5.2. The reason we restrict to the Frobenius-orientation case, is that in the general case, the rank-1 action $\mathcal{M}_1 \times \mathcal{X}_1 \rightarrow \mathcal{X}_1$ has two orbits [27]. In the case that p is inert in R , we could of course restrict to one of these orbits to recover the necessary base-transitivity axioms. It is then folklore that in dimension one, we can then do the membership testing by looking at the action of the orientation on the invariant differential. In higher dimension, this should be generalized to checking that the action of R on $H^0(A, \Omega_{A/k}^1)$ agrees with the action on $H^0(E_0^g, \Omega_{E_0^g/k})$ [30, Definition 4.8]⁶.

We will discuss to what extent $(\mathcal{M}', \mathcal{X}', \star)$ satisfies the axioms of an N -EGMA. However, before that, it is interesting to pause to consider what happens in dimension 1, in light of Proposition 3.5. As already mentioned, restricting the modules to rank 1 results in the group $\mathcal{C}(R)$, and the Hermitian structure makes no difference here, as projective modules of rank 1 all have a canonical unimodular Hermitian form (see Remark 5.3 below). Thus, we recover the free and transitive group action underlying many post-quantum primitives, such as CSIDH [8] (or more accurately, CSURF since we are working with R maximal [7]) and (qt)-PEGASIS [8, 11, 12].

Remark 5.3. In rank 1, the Hermitian structure adds nothing, as all rank-1 modules $\mathfrak{a}$ have a canonical unimodular Hermitian form given by

$$\begin{aligned} H_{\mathfrak{a}} : \mathfrak{a} \times \mathfrak{a} &\rightarrow R \\ (\alpha, \beta) &\rightarrow \alpha \bar{\beta} / n(\mathfrak{a}). \end{aligned}$$

However, in higher dimensions, Hermitian structures are arguably the most important part: Steinitz theorem (see Theorem 2.3) implies that the monoid $\mathcal{M} = \mathbb{N} \times \mathcal{C}(R)$ as sets, with the operation given by $(r, [\mathfrak{a}]) \otimes (s, [\mathfrak{b}]) = (rs, [\mathfrak{a}]^s [\mathfrak{b}]^r)$, which can be seen by distributivity of the tensor-product. Thus, if one forgets about Hermitian forms and polarizations, the skeleton $(\mathcal{M}/\mathcal{C}(R), \mathcal{X}/\sim_{\mathcal{C}(R)}, \bar{\star})$ looks particularly simple, e.g. there is only one set-element in dimension d for any d coprime to the order of $\mathcal{C}(R)$ (this uses Lemma 5.8 below). However, the

⁶ We thank Damien Robert for explaining us this result.

monoid $\mathcal{M}'$, and subsequently the skeleton $(\mathcal{M}'/Cl(R), \mathcal{X}'/\sim_{\mathcal{C}(R)}, \overline{\star})$ generally has a vastly more complicated structure.

In the lemma below, we essentially identify the algorithms missing for making the action above a full EGMA. We give a discussion of these algorithms in Section 5.4.

Lemma 5.4. *Suppose that, for every $d \in \mathbb{N}$, there exist PPT algorithms for membership and canonical representation in $\mathcal{X}'_d$, and for sampling from a distribution on $\mathcal{M}'_d$ that is statistically close to uniform. Then $(\mathcal{M}', \mathcal{X}', \star)$ is an EGMA.*

Proof. Rankwise finiteness, base-freeness, base-transitivity, and efficient computation of the action were established in Theorem 5.1. The monoid operation (which now literally corresponds to tensor products), membership testing and rank-1 inversion for $\mathcal{M}'$ can be computed from the pseudo-bases, see Theorem 2.4 and Theorem 2.5. Equality testing can be performed by taking the trace-form down to $\mathbb{Z}$, applying the algorithm by Haviv and Regev [19], and checking if any of the (polynomially bounded) list of isometries commutes with the R -action.

Finally, we can generate an origin $x_0 \in \mathcal{X}'_1$, by generating a supersingular elliptic curve with a primitive R -orientation, using an algorithm by Wesolowski [36, Lemma 4]. The assumed algorithms provide the remaining requirements of Definition 4.1. $\square$

Remark 5.5. The proof above shows an important point: while the algorithms are polynomial-time in fixed rank and dimension, they might be very costly in varying rank and dimension, and thus in practice, we want to stay in low rank for constructing protocols. For instance, the algorithm by Haviv and Regev has complexity $O((2r)^{O(2r)})$ in the rank (which is also the upper bound on the size of the list of returned isometries), using that R has rank 2 over $\mathbb{Z}$.

We also remark that if we temporarily allow R to be the ring of integers in a larger number field, we see that equality testing in $\mathcal{M}'$ for $R = \mathbb{Z}[\zeta_{2^n}]$ is in fact closely related to the hard problem underlying the signature scheme HAWK [4]. Since these lattice-isomorphism-problems are generally easier than for unstructured lattices, it is likely that there exists more efficient algorithms for equality testing in $\mathcal{M}'_d$ as well.

5.2 The quotient action used in MIKE

MIKE does not work with the entire action above [2]. Instead, we now show how MIKE uses purely the skeletal action from Remark 4.2, which gives the missing near-uniform sampling and membership testing in rank 2, as well as equality testing in rank 4.

For this subsection, we fix $p \equiv 3 \pmod{4}$, which implies $R = \mathbb{Z}[(1 + \sqrt{-p})/2]$, as in MIKE. In this case, $\mathcal{C}(R)$ is odd, and in light of Lemma 5.8, the actions of $\mathcal{C}(R)$ in even dimensions are free. Further, combining Lemma 5.8 and Steinitz

theorem shows that any $P \in (\mathcal{M}'/\mathcal{C}(R))_{2^d}$ has a unique representative with trivial Steinitz class.

This justifies the fact that in MIKE, we work with another monoid, $\mathcal{M}^0$, consisting of positive-definite unimodular Hermitian modules of trivial Steinitz class, and the corresponding set $\mathcal{X}^0$ given by the action of $\mathcal{M}^0$ on

$$E_0 : y^2 = x^3 - x,$$

where the R -action is given by Frobenius⁷. Since $\mathcal{X}_1^0 = \{[E_0]\}$, this is necessarily our choice of origin.

As the theorem below shows, this is still only enough to get a $\{1, 2\}$ -EGMA. However, the algorithms that are missing are not necessary to instantiate the corresponding key-exchange protocol based on an $\{1, 2, 4\}$ -EGMA (see Section 6.1).

Theorem 5.6. *Suppose that R is as above, and that there exist PPT algorithms for computing canonical representations and membership testing in $\mathcal{X}_4^0$ and sampling from a distribution on $\mathcal{M}_4^0$ that is statistically close to uniform. Then $(\mathcal{M}^0, \mathcal{X}^0, \star)$ is an $\{1, 2, 4\}$ -EGMA.*

Proof. As this is equivalent to the skeleton of the action in Lemma 5.4 (at the relevant ranks), all properties except the already missing canonical representations, membership testing and random sampling follow.

For our chosen parameters (where $\sqrt{-p}$ corresponds to Frobenius, and the class number is odd), a result by Robert and Page [31] shows that $\mathcal{X}'_2$ can be decomposed as the set

$$\left\{ \begin{array}{c} \text{isomorphism classes of supersingular elliptic curves} \\ \text{over } \mathbb{F}_{p^2}, \text{ up to Galois conjugacy} \end{array} \right\} \times \mathcal{C}(R).$$

The first component of the above product corresponds to $\mathcal{X}^0$ by Scholten construction, which give an easy membership test in $\mathcal{X}^0$. Further, on the algebraic side, this also shows that $\mathcal{M}_2$ can be identified as a set as

$$\mathcal{Cs}(\text{End}(E_0)) \times \mathcal{C}(R),$$

where the Steinitz class of the corresponding module corresponds to the element in the second component, giving a bijection $\mathcal{Cs}(\text{End}(E_0)) \leftrightarrow \mathcal{M}_2^0$, where the former admits well-known algorithms for statistically close to uniform random sampling. □

Note that even though the action $(\mathcal{M}^0, \mathcal{X}^0, \star)$ is only an $\{1, 2\}$ -EGMA, we have the necessary algorithms for instantiating the key-exchange and PKE in Section 6. Random sampling in $\mathcal{M}_4^0$ and membership-testing in $\mathcal{X}_4^0$ is not needed.

⁷ Note that this is not necessarily isomorphic to the quotient action, as in degrees where $\#\mathcal{C}(R)[d] > 1$, we can have that $\mathfrak{a} \otimes P \neq P$ for $\mathfrak{a} \in \mathcal{C}(R)[d]$ and $P \in \mathcal{M}_d^0$. However, since we will only look at things generated by $\mathcal{M}_2^0$, this is in fact equivalent.

And finally, even though efficiently computable invariants in $\mathcal{X}'_d$ for $d \geq 3$ is a big open problem, the shared keys in MIKE land in the special locus

$$\{((M_A, H_A) \otimes (M_B, H_B)) \star E_0 \mid M_A, M_B \text{ free}\} \subset \mathcal{X}_4^0,$$

for which the MIKE-paper shows how to compute invariants. A priori, this is not sufficient for canonical representation, since it seemingly only defines the shared secret up to R -conjugation, however the results of [16] show that these varieties are also invariant under R -conjugation.

Remark 5.7. In the interest of transparency, we emphasize that the proof of Theorem 5.6 relies on the unpublished work by Robert and Page. The fact that the Scholten's construction lie in $\mathcal{X}_2^0$ follows from results in [30], however, the missing part is that this is all of $\mathcal{X}_2^0$. For some intuition on this result, see the relevant conjecture by Robert [30, Remark 5.23].

5.3 Analysis of assumptions for the module-action.

Recall that in Figure 1, the action of M_1 is not necessarily free in every dimension. This is also true for the Hermitian module action $(\mathcal{M}', \mathcal{X}', \star)$. However, we have an easy criterion to determine when it is.

Lemma 5.8. *The group $\mathcal{C}(R) = \mathcal{M}_1$ acts freely on $\mathcal{X}'_d$ if and only if we have that $\mathcal{C}(R)[d] = \{[R]\}$.*

Proof. Let $(P, H) \in \mathcal{M}'_d$, and consider $(\mathfrak{a}, H_{\mathfrak{a}}) \in \mathcal{M}'_1$. If $\text{st}(P) = [\mathfrak{b}]$, we have that $\text{st}(\mathfrak{a} \otimes P) = [\mathfrak{a}^d \mathfrak{b}]$, and thus, we see that $[\mathfrak{a}^d \mathfrak{b}] = [\mathfrak{a}]^d [\mathfrak{b}] = [\mathfrak{b}]$ implies that $[\mathfrak{a}]$ has order dividing d , and hence it must be trivial.

For the converse, let $\mathfrak{a} \subset R$ be a non-principal ideal with $[\mathfrak{a}] \in \mathcal{C}(R)[d]$, and let $P = R \oplus \mathfrak{a} \oplus \dots \oplus \mathfrak{a}^{d-1}$ (with the product Hermitian form). Then we have

$$\mathfrak{a} \otimes P \simeq \mathfrak{a} \oplus \mathfrak{a}^2 \oplus \dots \oplus \mathfrak{a}^d \simeq \mathfrak{a} \oplus \mathfrak{a}^2 \oplus \dots \oplus R \simeq P$$

(even as Hermitian modules!), and thus the action cannot be free in dimension d . $\square$

Proposition 5.9. *For $\mathcal{M}$, there exists an algorithm for $(r, 2)$ -extraction. Assume now that $\#\mathcal{C}(R)[2] \in O(1)$, then there exists an algorithm for full $(r, 2)$ -extraction. Under the same assumption, there exists an algorithm for full $(1, 2)$ -extraction for $\mathcal{M}'$.*

Proof. Assume we are given the instance $P_r \otimes P_2 =: P_{2r} \in \mathcal{M}_{2r}$, where P_2 and P_{2r} are public, and $P_r \in \mathcal{M}_r$ is secret. Computing steinitz classes gives

$$\text{St}(P_{2r}) = \text{St}(P_r \otimes P_2) = \text{St}(P_r)^2 \text{St}(P_2)^r,$$

i.e.

$$\text{St}(P_r)^2 = \text{St}(P_{2r}) \text{St}(P_2)^{-r}.$$

By dividing by 2 in the class group, which can be done efficiently, even without knowing the class number [5], we reveal an element $[\mathfrak{a}] \in \mathcal{C}(R)$ with $[\mathfrak{a}]^2 = \text{St}(P_r)^2$. For $\mathcal{M}$, this is sufficient, as we can take a solution as $R^{r-1} \otimes [\mathfrak{a}]$. Further, it is easy to see that all possible solutions are in bijection with $\mathcal{C}(R)[2]$ (which is efficiently enumerable), by the map $[\mathfrak{b}] \rightarrow [\mathfrak{b}\mathfrak{a}]$, where $[\mathfrak{b}] \in \mathcal{C}(R)[2]$. Thus, the assumption gives full $(r, 2)$ -extraction.

If further $r = 1$, the above algorithm also reveals P_r as a Hermitian module (see Remark 5.3), hence solving (full) $(1, 2)$ -extraction for $\mathcal{M}'$. $\square$

We leave it as an interesting open question whether the above proposition can be extended to solve $(r, 2)$ -extraction for $(\mathcal{M}', \mathcal{X}', \star)$ for any $r \in \mathbb{N}$.

Finally, we restate an important result due to Robert [30, Section 5.4], in the context of an EGMA.

Theorem 5.10. *The supersingular endomorphism-ring problem reduces to the $(1, 2)$ -graded vectorization problem for $(\mathcal{M}^0, \mathcal{X}^0, \star)$.*

Proof. This is exactly the statement and proof from [30, Theorem 5.22], in our language: given a supersingular elliptic curve $E/\mathbb{F}_{p^2}$, form an $(1, 2)$ -graded vectorization instance from $E_0 \in \mathcal{X}_1^0$ and $W'_{\mathbb{F}_{p^2}/\mathbb{F}_p}(E) \in \mathcal{X}_2^0$. This returns a free module $(M, H) \in \mathcal{M}_2^0$ such that $W'_{\mathbb{F}_{p^2}/\mathbb{F}_p}(E) = (M, H) \star E_0$, from which we can recover a left quaternion ideal $I \subseteq \text{End}(E_0)$ [30, Theorem 5.5] such that $(I, \text{trd}_{\text{End}(E_0)/R} \circ H_I) \simeq (M, H)$ as R -lattices, which reveals $\text{End}(E) = \mathcal{O}_R(I)$ (here, H_I is the canonical Hermitian form of I as an $\text{End}(E_0)$ -lattice). $\square$

5.4 Open problems towards a full EGMA

The results above reveals some necessary algorithms for $(\mathcal{M}', \mathcal{X}', \star)$ to be a full N-EGMA. In light of Lemma 5.4, the main missing algorithms are the following:

- *Near-uniform sampling.* For every fixed rank d , construct a PPT algorithm that samples positive-definite unimodular Hermitian R -modules from a distribution statistically close to uniform on $\mathcal{M}'_d$. By sampling pseudo-bases and suitable Gram-matrices directly, one obtains easy algorithms for *Heuristic* near-uniform sampling, but we are not aware of any provable algorithms.
- *Membership testing.* For every fixed rank and dimension, efficiently recognize valid representations of elements of $\mathcal{X}'_d$. For the MIKE action $(\mathcal{M}^0, \mathcal{X}^0, \star)$ to be an $\{1, 2, 4\}$ -EGMA, the remaining case is near uniform sampling in rank 4, and membership testing $\mathcal{X}'_4$, though we again stress that this is not necessary to instantiate the key-exchange in Section 6.1 itself.
- *Canonical representations.* For every fixed rank and dimension, efficiently compute canonical representations of elements in $\mathcal{X}'_d$. General-purpose efficiently computable invariants are not currently known for the set side in dimension at least 3. The MIKE invariants solve this problem only for the special family of abelian fourfolds occurring in MIKE.

We leave the question of whether these problems admit efficient algorithms as an interesting open problem.

6 Constructions

In this section, we give a few simple constructions, to illustrate how one can use the EGMA-framework to construct protocols. The first is a family of simple key-exchanges, which can be instantiated with $(\mathcal{M}^0, \mathcal{X}^0, \star)$, resulting in the MIKE-protocol.

Second, we give a family of novel sigma-protocol with binary challenges, which may also be instantiated with $(\mathcal{M}^0, \mathcal{X}^0, \star)$. However, we emphasize that its security relies on the assumption of existence of an efficient algorithm for $(2, 2)$ -extraction.

All our protocols are direct generalizations of the corresponding protocols based on class-group actions in dimension 1, in the sense that the families we define also cover these protocols.

6.1 Families of EGMA-based non-interactive key exchanges and PKEs

The first protocol is a simple Diffie-Hellman like protocol. It requires an $\{1, r, s, rs\}$ -EGMA, although we will remark that this can be relaxed, as we do not need all algorithms in rank and dimension rs (see Remark 6.1).

Given a $\{1, r, s, rs\}$ -EGMA $(M, X, \star)$, we define the (r, s) -DH protocol by the following algorithms (see Figure 3):

- $\text{Pgen}(\lambda)$: On the input of a security parameter λ generates public parameters, an instantiation of $\text{pp} := (M, X, \star)$ at the suitable security level. In the following algorithms, pp will be used as input, even if we do not explicitly write it.
- $\text{KGen}(t)$: On input $t \in \{r, s\}$, samples random secret element $\text{sk} := a \in M_t$ (according to the distribution χ_t) and computes the public key $\text{pk} := a \star x_0 \in X_t$. The function outputs (pk, sk) .
- $\text{KA}(\text{pk}_2, \text{sk}_1)$: On input a public key $\text{pk}_2 \in X_r$ and secret key $\text{sk}_1 \in M_s$ (or vice versa with respect to the rank and dimension), computes the shared key $\text{sk}_1 \star \text{pk}_2 \in M_{rs}$.

Of course, the security of the above protocol clearly relies on the hardness of $(1, r, s)$ -graded vectorization for $(M, X, \star)$. However, when the protocol is instantiated with $(\mathcal{M}^0, \mathcal{X}^0, \star)$ and $r = s = 2$, the protocol can be proven secure in a suitable model, and assuming a natural gap-assumption, as shown in the MIKE paper [2].

Remark 6.1. Clearly, a full $\{1, r, s, rs\}$ -EGMA is not necessary, as the protocol above never uses random sampling in M_{rs} , nor membership testing in X_{rs} . Canonical representations in X_{rs} is seemingly necessary, however, as in MIKE, it is also sufficient to have canonical representations of the key-space, which in general is a strict subset of X_{rs} .

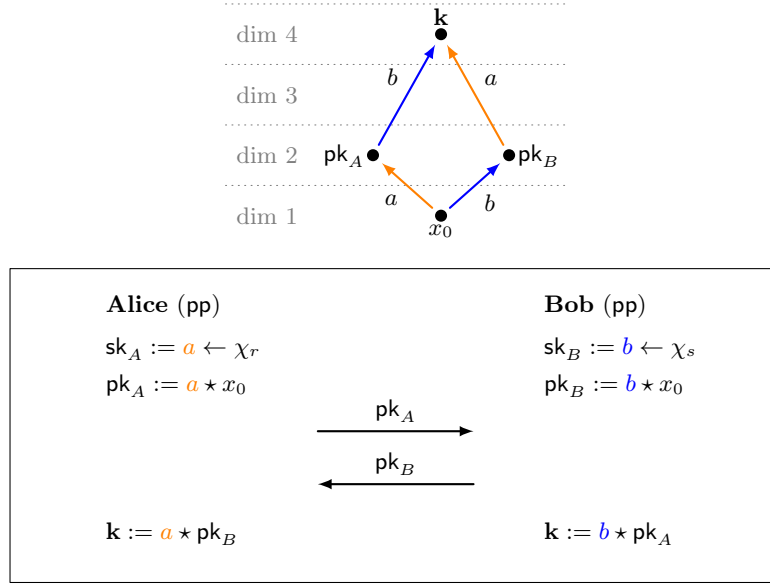


Fig. 3. A non-interactive key-exchange protocol using module action. The upper diagram illustrates the graded relations for $r = s = 2$, while the lower diagram shows the protocol for general r and s .

Remark 6.2. Note that if the protocol above is instantiated with $(\mathcal{M}', \mathcal{X}', \star)$, which by (qt)-PEGASIS is a $\{1\}$ -EGMA [11, 12], and $r = s = 1$, we recover the (1, 1)-DH protocol as in the original CRS-key exchange. Of course, this then only gives quantum-subexponential security, which is why the (2, 2)-DH protocol for $(\mathcal{M}^0, \mathcal{X}^0, \star)$, whose security is (conjecturally) exponential, massively outperforms the $r = s = 1$ version, both with respect to key-sizes and computational time. Concretely, the implementation of MIKE reports timings as low as 7 ms, with public-key sizes at 80 bytes for NIST level 1.

The same construction can of course be adapted to public-key encryption [32]. The sender computes (a hash of) the canonical representation of the shared secret to obtain a bit string, which is XORed with the message. This follows the same structure of the hashed version of the famous ElGamal [15] scheme.

Given a $\{1, r, s, rs\}$ -EGMA $(M, X, \star)$, the (r, s) -PKE scheme reuses $\text{Pgen}(\lambda)$ and $\text{KGen}(r)$ from the key-exchange protocol to generate the public parameters and the receiver's key pair $(pk, sk) := (b \star x_0, b)$. Fix a message space $\mathcal{M} = \{0, 1\}^\ell$ and a hash function $H : X_{rs} \rightarrow \mathcal{M}$ based on the canonical representation, included in the public parameters. As before, these parameters are implicit inputs to the algorithms below. Writing xor for bitwise XOR, encryption and decryption proceed as follows (see Figure 4):

- $\text{Enc}(\text{pk}, m)$: On input a public key $\text{pk} \in X_r$ and a message $m \in \mathcal{M}$, samples $b \leftarrow \chi_s$ and computes $c_1 := b \star x_0 \in X_s$ and the shared value $\mathbf{k} := b \star \text{pk} \in X_{rs}$. It sets $c_2 := \text{H}(\mathbf{k}) \text{ xor } m$ and outputs the ciphertext $c := (c_1, c_2)$.
- $\text{Dec}(\text{sk}, c)$: On input a secret key $\text{sk} := a \in M_r$ and a ciphertext $c = (c_1, c_2)$, computes $\mathbf{k} := a \star c_1$ and outputs $m := \text{H}(\mathbf{k}) \text{ xor } c_2$.

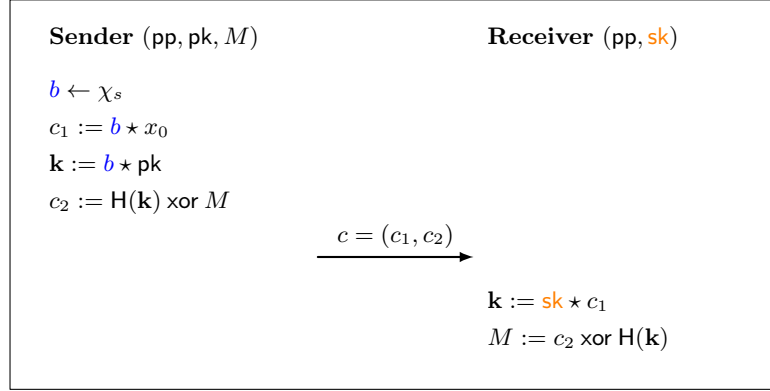


Fig. 4. A PKE protocol using module action, showing encryption and decryption for general r and s .

Any adversary which can break the $(1, r)$ -graded vectorization problem (4.3), can obtain the secret key from the public key since $\text{pk} = a \star x_0$, and any adversary which can break the $(1, r, s)$ -graded parallelization problem (4.4) can obtain the ephemeral key $\mathbf{k} = a \star (b \star x_0)$ from $\text{pk} = a \star x_0, c_1 = b \star x_0$ and compute $M = \text{H}(\mathbf{k}) \text{ xor } c_2$.

ElGamal is known to be IND-CPA secure if the CDH problem is hard in the random oracle model. The approach in [18, Theorem 7.1.0.1] is easily adapted to showing the same for our scheme by simply swapping g^a, g^b, g^{ab} for $a \star x_0, b \star x_0, a \star (b \star x_0)$, as well as swapping CDH for the $(1, r, s)$ -graded parallelization problem. Essentially, if an adversary can distinguish between a ciphertext $(c_1, c_2) = (b \star x_0, \text{H}(b \star (a \star x_0) \text{ xor } m))$ when m is randomly selected from one of two messages m_0, m_1 chosen by the adversary, and H is a random oracle, they will have demonstrated that they have knowledge of the input value to the hash function H . The value c_1 is independent of the message. Instantiated as a $(1, 2, 2)$ -protocol, MIKE [2] provides this security.

6.2 Families of EGMA-based sigma protocols

We now go on to define a family of EGMA-based sigma protocols. Unlike the case of key-exchange, we will see that these protocols require assumptions not provided by [2].

Our (family of) protocol(s) is constructed as the natural generalization of the graph-isomorphism protocol underlying CSI-FiSh [3] (see Remark 6.6). The same protocol has also recently been considered in the context of finite semi-group actions [17]. Although the context is quite different than the EGMA-setting, we also encounter the same issues as in [17]: namely the fact that to get special soundness, we need an additional assumption. We will later see that for large families of protocols, instantiated as in Section 5, this assumption is satisfied. Unfortunately, for the families of protocols we can prove secure, although they generalize the corresponding EGA-based protocols, they seem to offer no particular benefit. It is an open problem whether we can find a secure instantiation, for which there are no sub-exponential attacks.

We define the basic (r, s) -ID protocol as follows. Given an $\{1, r, s, rs\}$ -EGMA, the protocols runs $\text{Pgen}(\lambda)$ from before, fixing the public parameters, and the signer generates their private key by $\text{sk}, \text{pk} \leftarrow \text{KGen}(r)$, where $\text{sk} := a \in M_r$, and $\text{pk} := \text{sk} \star x_0$. The protocol then proceeds as by the following algorithms (see Figure 5).

- **Commit(pk)**: Sample a random $c \in M_s$ (according to the distribution χ_s), and compute $y_{\text{com}} := c \star \text{pk}$. The algorithm outputs (c, y_{com}) .
- **Chal**: Sample a random bit $b \in \{0, 1\}$, and output b .
- **Resp(sk, c, y_{com}, b)**: On input a secret key $\text{sk} \in M_r$, the commitment pair (c, y_{com}) , and a challenge bit b , set $z := c$ if $b = 0$, or compute $z := c \otimes a$ if $b = 1$. The algorithm then outputs z .
- **Verify(pk, y_{com}, b, z)**: On input a public key pk , a public commitment y_{com} , the challenge b , and the response z , the function checks that $z \star \text{pk} = y_{\text{com}}$ if $b = 0$, or that $z \star x_0 = y_{\text{com}}$ if $b = 1$. Output **True** if this check passes, and **False** otherwise.

Note that we can again instantiate the $(2, 2)$ -ID protocol with $(\mathcal{M}^0, \mathcal{X}^0, \star)$, since Remark 6.1 still holds. However, as the following theorem specifies, its security would rely on additional assumptions related to (r, s) -hiding and (r, s) -extraction.

Theorem 6.3. *Let $r, s \in \mathbb{N}$, and let $N = \{1, r, s, rs\}$. Given an N -EGMA $(M, X, \star)$, for which $(1, r)$ -graded vectorization is hard, the (r, s) -ID protocol defined above satisfies perfect completeness and is computational honest-verifier zero knowledge assuming the hardness of (r, s) -hiding. Moreover, if there is an efficient algorithm for full (r, s) -extraction, then the protocol is special sound.*

Proof. Completeness is immediate. To simulate a transcript with challenge $b = 0$, sample $c \leftarrow \chi_s$ and output the commitment $c \star \text{pk}$ and response c . This is distributed exactly as an honest transcript. For $b = 1$, sample $a' \leftarrow \chi_r$ and $c' \leftarrow \chi_s$, set $z = a' \otimes b'$, and output the commitment $z \star x_0$ and response z . By the hardness of (r, s) -hiding, the joint distributions of the public key and response in the simulated and honest transcripts are computationally indistinguishable. The same therefore holds after applying the action to form the commitment.

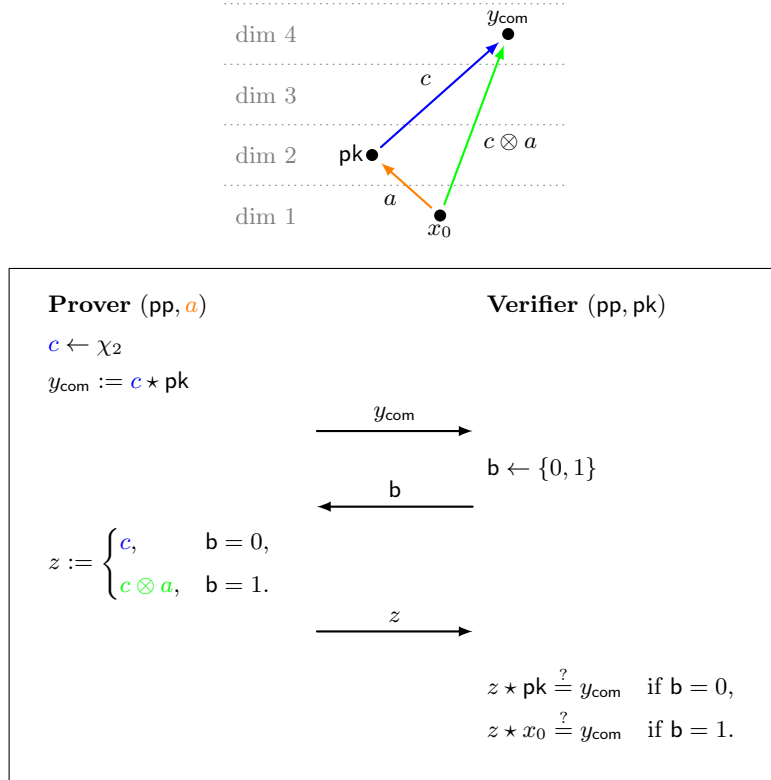


Fig. 5. An illustration of the (2,2)-ID protocol. The upper diagram shows the graded relations used during verification, while the lower diagram shows the three-message protocol and the verifier's checks.

For special soundness, suppose that two accepting transcripts have the same commitment and responses $z_0 \in M_s$ and $z_1 \in M_{rs}$ to the challenges 0 and 1, respectively. Full (r, s) -extraction gives all elements of the fibre $(z_0 \otimes -)^{-1}(z_1)$, from which we can output the unique element $a \in M_r$ satisfying $a \star x = \text{pk}$, where uniqueness is implied by base-freeness, which gives a solution to the $(1, 2)$ -graded vectorization problem (pk, x_0) . $\square$

Remark 6.4. A potential source of confusion in the theorem above, is that at first glance, the proof does not seem to rely on (r, s) or $(1, sr)$ -graded vectorization being hard. However, this assumption is already implied by the hardness of (r, s) -hiding, full (r, s) -extraction and $(1, r)$ -graded vectorization. The proof of this is essentially in the proof of the protocol: As an example, if (r, s) -graded vectorization was easy, then the adversary could take the $(1, r)$ -graded vectorization instance (pk, x_0) , and sample random $a' \in M_r$ and $b' \in M_s$, and attempt to solve the (r, s) -graded vectorization instance $(\text{pk}, (a' \otimes b') \star x_0)$. If such an instance have a non-negligible chance of having no solutions, then (r, s) -hiding cannot be hard when (r, s) -graded vectorization is easy, while if there is a solution, the adversary recovers the solution to the $(1, r)$ -graded vectorization instance by the extraction argument. A similar reduction occurs from an efficient algorithm for $(1, rs)$ -graded vectorization.

Corollary 6.5. *For any $r \in \mathbb{N}$, the $(r, 1)$ -ID protocol has special soundness assuming the hardness of $(r, 1)$ -graded vectorization, and $(r, 1)$ -hiding.*

Proof. By the discussion at the end of Section 5, full $(r, 1)$ -extraction is trivial by rank-1 inversion. $\square$

Remark 6.6. We again remark that if the $(1, 1)$ -ID protocol is instantiated with $(\mathcal{M}', \mathcal{X}', \star)$, we recover the original CRS-based ID-protocol [10, 33, 3]. It is interesting that the usual group-action-based ID protocol generalizes to certain instances in higher dimension, though we stress that there seems to be no particular benefit to the $(r, 1)$ -ID protocol over the simple $(1, 1)$ -ID protocol, since its secret still has rank 1, and security is thus only quantum sub-exponential. In contrast, the $(1, 2)$ -ID protocol has a rank-2 secret, but since recovering the commitment is a $(2, 1)$ -graded vectorization problem, we still only get sub-exponential security. Further, it requires the additional assumption that $(2, 1)$ -hiding is hard.

However, an efficient algorithm solving full $(2, 2)$ -extraction for $(\mathcal{M}', \mathcal{X}', \star)$ would allow us to prove the corresponding $(2, 2)$ -ID protocol secure, under the assumption that $(2, 2)$ -hiding is hard. Its witness relation is again $(1, 2)$ -graded vectorization, which reduces to the endomorphism-ring problem Theorem 5.10.

References

- [1] N. Alapati, L. De Feo, H. Montgomery, and S. Patranabis. “Cryptographic Group Actions and Applications”. In: *ASIACRYPT 2020, Part II*. Ed. by S. Moriai and H. Wang. Vol. 12492. LNCS. Springer, Cham, Dec. 2020, pp. 411–439. DOI: [10.1007/978-3-030-64834-3_14](https://doi.org/10.1007/978-3-030-64834-3_14).

- [2] A. Basso, P. Dartois, M. Duparc, J. K. Eriksen, S. Kunzweiler, M. Meyer, G. Pope, K. Reijnders, D. Robert, R. Rueger, and S. Schaeffler. *MIKE: A Fast and Compact Post-Quantum NIKE*. Cryptology ePrint Archive, to appear. 2026.
- [3] W. Beullens, T. Kleinjung, and F. Vercauteren. “CSI-FiSh: Efficient Isogeny Based Signatures Through Class Group Computations”. In: *ASIACRYPT 2019, Part I*. Ed. by S. D. Galbraith and S. Moriai. Vol. 11921. LNCS. Springer, Cham, Dec. 2019, pp. 227–247. DOI: [10.1007/978-3-030-34578-5_9](https://doi.org/10.1007/978-3-030-34578-5_9).
- [4] J. W. Bos, O. Bronchain, L. Ducas, S. Fehr, Y. Huang, T. Pornin, E. W. Postlethwaite, T. Prest, L. N. Pulles, and W. van Woerden. *HAWK*. Tech. rep. available at <https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures>. National Institute of Standards and Technology, 2024.
- [5] W. Bosma and P. Stevenhagen. “On the computation of quadratic 2-class groups”. In: *Journal de théorie des nombres de Bordeaux* 8.2 (1996), pp. 283–313.
- [6] G. Brassard and M. Yung. “One-Way Group Actions”. In: *CRYPTO’90*. Ed. by A. J. Menezes and S. A. Vanstone. Vol. 537. LNCS. Springer, Berlin, Heidelberg, Aug. 1991, pp. 94–107. DOI: [10.1007/3-540-38424-3_7](https://doi.org/10.1007/3-540-38424-3_7).
- [7] W. Castryck and T. Decru. “CSIDH on the Surface”. In: *PQCrypto 2020*. Ed. by J. Ding and J.-P. Tillich. Springer, Cham, 2020, pp. 111–129. DOI: [10.1007/978-3-030-44223-1_7](https://doi.org/10.1007/978-3-030-44223-1_7).
- [8] W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes. “CSIDH: An Efficient Post-Quantum Commutative Group Action”. In: *ASIACRYPT 2018, Part III*. Ed. by T. Peyrin and S. Galbraith. Vol. 11274. LNCS. Springer, Cham, Dec. 2018, pp. 395–427. DOI: [10.1007/978-3-030-03332-3_15](https://doi.org/10.1007/978-3-030-03332-3_15).
- [9] H. Cohen. *Advanced topics in computational number theory*. Springer Science & Business Media, 2012.
- [10] J.-M. Couveignes. *Hard Homogeneous Spaces*. Cryptology ePrint Archive, Report 2006/291. 2006. URL: <https://eprint.iacr.org/2006/291>.
- [11] P. Dartois, J. K. Eriksen, T. B. Fouotsa, A. H. L. Merdy, R. Invernizzi, D. Robert, R. Rueger, F. Vercauteren, and B. Wesolowski. “PEGASIS: Practical Effective Class Group Action using 4-Dimensional Isogenies”. In: *CRYPTO 2025, Part I*. Ed. by Y. T. Kalai and S. F. Kamara. Vol. 16000. LNCS. Springer, Cham, Aug. 2025, pp. 67–99. DOI: [10.1007/978-3-032-01855-7_3](https://doi.org/10.1007/978-3-032-01855-7_3).
- [12] P. Dartois, J. K. Eriksen, R. Invernizzi, and F. Vercauteren. *qt-Pegasis: Simpler and Faster Effective Class Group Actions*. Cryptology ePrint Archive, Report 2025/1859. 2025. URL: <https://eprint.iacr.org/2025/1859>.
- [13] L. De Feo, D. Kohel, A. Leroux, C. Petit, and B. Wesolowski. “SQISign: Compact Post-quantum Signatures from Quaternions and Isogenies”. In: *ASIACRYPT 2020, Part I*. Ed. by S. Moriai and H. Wang. Vol. 12491. LNCS. Springer, Cham, Dec. 2020, pp. 64–93. DOI: [10.1007/978-3-030-64837-4_3](https://doi.org/10.1007/978-3-030-64837-4_3).

- [14] W. Diffie and M. E. Hellman. “New Directions in Cryptography”. In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: [10.1109/TIT.1976.1055638](https://doi.org/10.1109/TIT.1976.1055638).
- [15] T. ElGamal. “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”. In: *CRYPTO’84*. Ed. by G. R. Blakley and D. Chaum. Vol. 196. LNCS. Springer, Berlin, Heidelberg, Aug. 1984, pp. 10–18. DOI: [10.1007/3-540-39568-7_2](https://doi.org/10.1007/3-540-39568-7_2).
- [16] J. K. Eriksen, R. Invernizzi, A. Sferlazza, P. Tasbihgou, and F. Vercauteren. *Cryptanalysis of Higher-Dimensional Module Actions via Twisting and Genus Theory*. Cryptology ePrint Archive, to appear. 2026.
- [17] O. W. Gnilke and J. Zumbärgel. *Cryptographic Group and Semigroup Actions*. Cryptology ePrint Archive, Report 2023/017. 2023. URL: <https://eprint.iacr.org/2023/017>.
- [18] F. Guo, W. Susilo, and Y. Mu. “Public-Key Encryption with Random Oracles”. In: *Introduction to Security Reduction*. Cham: Springer International Publishing, 2018, pp. 193–207. ISBN: 978-3-319-93049-7. DOI: [10.1007/978-3-319-93049-7_7](https://doi.org/10.1007/978-3-319-93049-7_7). URL: https://doi.org/10.1007/978-3-319-93049-7_7.
- [19] I. Haviv and O. Regev. “On the Lattice Isomorphism Problem”. In: *25th SODA*. Ed. by C. Chekuri. ACM-SIAM, Jan. 2014, pp. 391–404. DOI: [10.1137/1.9781611973402.29](https://doi.org/10.1137/1.9781611973402.29).
- [20] M. Kirschmer. “Definite quadratic and hermitian forms with small class number”. In: *Habilitation, RWTH Aachen University* (2016), p. 13.
- [21] N. Koblitz. “Constructing Elliptic Curve Cryptosystems in Characteristic 2”. In: *Advances in Cryptology - CRYPTO ’90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*. Ed. by A. Menezes and S. A. Vanstone. Vol. 537. Lecture Notes in Computer Science. Springer, 1990, pp. 156–167. DOI: [10.1007/3-540-38424-3_11](https://doi.org/10.1007/3-540-38424-3_11). URL: https://doi.org/10.1007/3-540-38424-3_11.
- [22] P. Kotzias, A. Razaghpanah, J. Amann, K. G. Paterson, N. Vallina-Rodriguez, and J. Caballero. “Coming of Age: A Longitudinal Study of TLS Deployment”. In: *Proceedings of the Internet Measurement Conference 2018, IMC 2018, Boston, MA, USA, October 31 - November 02, 2018*. ACM, 2018, pp. 415–428. URL: <https://dl.acm.org/citation.cfm?id=3278568>.
- [23] G. Kuperberg. “A Subexponential-Time Quantum Algorithm for the Dihedral Hidden Subgroup Problem”. In: *SIAM J. Comput.* 35.1 (2005), pp. 170–188. DOI: [10.1137/S0097539703436345](https://doi.org/10.1137/S0097539703436345). URL: <https://doi.org/10.1137/S0097539703436345>.
- [24] V. S. Miller. “Use of Elliptic Curves in Cryptography”. In: *CRYPTO’85*. Ed. by H. C. Williams. Vol. 218. LNCS. Springer, Berlin, Heidelberg, Aug. 1986, pp. 417–426. DOI: [10.1007/3-540-39799-X_31](https://doi.org/10.1007/3-540-39799-X_31).
- [25] J. S. Milne. “Abelian varieties”. In: *Arithmetic geometry* (1986), pp. 103–150.

- [26] A. M. Odlyzko. “Discrete Logarithms in Finite Fields and Their Cryptographic Significance”. In: *EUROCRYPT’84*. Ed. by T. Beth, N. Cot, and I. Ingemarsson. Vol. 209. LNCS. Springer, Berlin, Heidelberg, Apr. 1985, pp. 224–314. DOI: [10.1007/3-540-39757-4_20](https://doi.org/10.1007/3-540-39757-4_20).
- [27] H. Onuki. “On oriented supersingular elliptic curves”. In: *Finite Fields and Their Applications* 69 (2021), p. 101777.
- [28] A. Page and D. Robert. *Introducing Clapoti(s): Evaluating the isogeny class group action in polynomial time*. Cryptology ePrint Archive, Report 2023/1766. 2023. URL: <https://eprint.iacr.org/2023/1766>.
- [29] A. Page, D. Robert, and J. Soumier. “On the conversion of module representations for higher dimensional supersingular isogenies”. In: *IACR Cryptol. ePrint Arch.* 2026 (2026), p. 276. URL: <https://eprint.iacr.org/2026/276>.
- [30] D. Robert. *The module action for isogeny based cryptography*. Cryptology ePrint Archive, Report 2024/1556. 2024. URL: <https://eprint.iacr.org/2024/1556>.
- [31] D. Robert and A. Page. *Clapotis*. Cryptology ePrint Archive, to appear.
- [32] J. Rosenthal and S. Sconza. *Semigroup Action Problems and Their Uses in Post-Quantum Cryptography*. Cryptology ePrint Archive, Paper 2026/462. 2026. URL: <https://eprint.iacr.org/2026/462>.
- [33] A. Rostovtsev and A. Stolbunov. “Public-Key Cryptosystem Based on Isogenies”. In: *IACR Cryptol. ePrint Arch.* 2006 (2006), p. 145. URL: <http://eprint.iacr.org/2006/145>.
- [34] P. W. Shor. “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”. In: *35th FOCS*. IEEE Computer Society Press, Nov. 1994, pp. 124–134. DOI: [10.1109/SFCS.1994.365700](https://doi.org/10.1109/SFCS.1994.365700).
- [35] C. A. Weibel. *The K-book: An Introduction to Algebraic K-theory*. Vol. 145. American Mathematical Soc., 2013.
- [36] B. Wesolowski. “Orientations and the Supersingular Endomorphism Ring Problem”. In: *EUROCRYPT 2022, Part III*. Ed. by O. Dunkelman and S. Dziembowski. Vol. 13277. LNCS. Springer, Cham, May 2022, pp. 345–371. DOI: [10.1007/978-3-031-07082-2_13](https://doi.org/10.1007/978-3-031-07082-2_13).

A AI Disclosure

The ideas and contributions in this paper are all the authors’. AI tools (specifically gpt-5.6 and gpt-6) were used towards the end of the project for light editing, as well as the construction of Figure 3, Figure 4 and Figure 5, under human assistance. The authors take full responsibility for the correctness of all material in this work.