







Diffie–Hellman Key Exchange from Commutativity to Group Laws

Dung Hoang Duong  

Institute of Cybersecurity and Cryptology, School of Computing and Information Technology,
University of Wollongong, Australia

Youming Qiao  

School of Computer Science and Engineering, University of New South Wales, Sydney, Australia
Centre for Quantum Software and Information, University of Technology Sydney, Australia

Chuanqi Zhang  

Centre for Quantum Software and Information, University of Technology Sydney, Australia

Abstract

In Diffie–Hellman key exchange, the commutativity of power operations is instrumental in the agreement of keys. Viewing commutativity as a law in abelian groups, we propose Diffie–Hellman key exchange in the group action framework (Brassard–Yung, *Crypto*'90; Ji–Qiao–Song–Yun, *TCC*'19), for actions of non-abelian *groups with laws*. The security of this protocol is shown, following Fischlin, Günther, Schmidt, and Warinschi (*IEEE S&P*'16), based on a pseudorandom group action assumption. A concrete instantiation is proposed based on the monomial code equivalence problem.

2012 ACM Subject Classification Security and privacy → Mathematical foundations of cryptography

Keywords and phrases Diffie–Hellman, Key Exchange, Group Laws, Group Actions, Code Equivalence

Digital Object Identifier 10.4230/LIPIcs.ITCS.2026.52

Related Version *Full Version*: <https://eprint.iacr.org/2025/1677> [44]

Funding *Dung Hoang Duong*: Research supported in part by Australian Research Council LP220100332.

Youming Qiao: Research supported in part by Australian Research Council DP200100950 and LP220100332.

Chuanqi Zhang: Research supported in part by Australian Research Council DP200100950 and LP220100332, and the Sydney Quantum Academy, Sydney, NSW, Australia.

Acknowledgements We thank the anonymous reviewers for their insightful feedback.

1 Introduction

1.1 Background

Diffie–Hellman key exchange and commutativity. The celebrated Diffie–Hellman key exchange [39] is a public-key protocol allowing two parties to agree on a common key. This protocol is of significant historical importance and is widely used in practice.

Let us quickly review this protocol. For a prime p , let C_p be a cyclic group of order p and $\gamma \in C_p$ a generator of C_p . Alice (resp. Bob) randomly sample $a \in [p-1] := \{1, \dots, p-1\}$ (resp. $b \in [p-1]$). Alice computes γ^a and sends it to Bob. Bob computes γ^b and sends it to Alice. Alice computes $(\gamma^b)^a = \gamma^{ba}$ as her key. Bob computes $(\gamma^a)^b = \gamma^{ab}$ as his key.



© Dung Hoang Duong, Youming Qiao, and Chuanqi Zhang;
licensed under Creative Commons License CC-BY 4.0

17th Innovations in Theoretical Computer Science Conference (ITCS 2026).

Editor: Shubhangi Saraf; Article No. 52; pp. 52:1–52:20



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

It is readily seen that the *commutativity* of $ab = ba$ ensures that $\gamma^{ba} = \gamma^{ab}$ so Alice and Bob share the same key. The role of commutativity is natural and crucial, so various attempts to generalize Diffie–Hellman tried to exploit commutativity in one way or another, even when dealing with non-commutative objects [70, 72].

An intriguing question is then whether commutativity could be generalized or relaxed for reaching key agreement in the Diffie–Hellman-type key exchange protocols. Our goal in this article is to propose a natural generalization via *group laws* in the framework of *group action based cryptography* [26, 58].

Group action based cryptography. Cryptography based on group actions was first studied as a framework by Brassard and Yung [26], who proposed the definition of one-way group actions and presented some cryptographic applications. Couveignes developed this framework further but with a focus on commutative groups in [35], and one contribution of Couveignes is the proposal of using class group actions on elliptic curves in cryptography.

Diffie–Hellman key exchange can be formulated via group actions, as observed by Brassard and Yung [26] and Couveignes [35]. Using the notation in the description above, define the set S to be $C_p \setminus \{\text{id}\}$, the cyclic group C_p excluding the identity element id . The group $G := \text{Aut}(C_p)$ is the automorphism group of C_p . Note that G is isomorphic to $(\mathbb{Z}/p\mathbb{Z})^*$, the multiplicative group of $\mathbb{Z}/p\mathbb{Z}$, so we can identify $a \in G$ as an element in $(\mathbb{Z}/p\mathbb{Z})^*$. The group action is then $a \in (\mathbb{Z}/p\mathbb{Z})^*$ sending $\delta \in S = C_p \setminus \{\text{id}\}$ to δ^a .

Recently, two works [58, 2] further developed this framework by introducing new notions such as pseudorandom group actions (which naturally generalizes the Decisional Diffie–Hellman assumption [21]) and devising more cryptographic applications. In addition, more candidate group actions suitable for cryptographic uses were proposed. In [58], the general linear group action on tensors was suggested as a candidate for pseudorandom group actions. In [2], the group actions are class group actions with variations such as in [31, 17].

A major application of group action based cryptography is digital signatures. This construction is based on the Goldreich–Micali–Wigderson (GMW) zero-knowledge protocol for graph isomorphism [50] and the Fiat–Shamir (FS) transformation [48]. This GMW-FS construction has seen a recent revival as evidenced by the works [18, 34, 81] for actions by symmetric and general linear groups, and the works [17, 46] for class group actions. See also [22, 19, 14] for recent progress and surveys.

Concrete group actions used in cryptography. We recall some concrete group actions proposed for use in cryptography, besides the group action underlying discrete logarithm.

On the commutative (abelian) group side, a main candidate is the action of the ideal-class group $\text{cl}(\mathcal{O})$ on the set of \mathbb{F}_q -isomorphism classes of ordinary elliptic curves over \mathbb{F}_q whose endomorphism ring is a given order \mathcal{O} in an imaginary quadratic field. It was later adapted by Castryck et al. [31] to the case of supersingular elliptic curves resulting in an efficient key exchange protocol called CSIDH. Based on CSIDH, many isogeny-based cryptographic constructions are built, such as digital signatures SeaSign [46] and CSI-FiSh [17], threshold signature [47], ring signatures [16], group signatures [15], blind signatures [59], updatable encryption [63], and password-authenticated key exchange (PAKE) [1, 57].

On the non-commutative (non-abelian) group side, there are three problem families, namely linear code equivalence [13, 18], tensor isomorphism and its variations [58, 51, 34, 81], and lattice isomorphism [53, 11, 42]. These are based on actions by monomial groups, general linear groups over finite fields, and general linear groups over \mathbb{Z} . Four digital signature schemes, LESS [18], MEDS [34], ALTEQ [20], and Hawk [42], were submitted to the NIST’s call for post-quantum digital signature schemes, with LESS and Hawk making into round 2.

Comparisons of some cryptographic commutative and non-commutative group actions.

When groups are commutative, then key exchange protocols and public-key encryptions can be realized following Diffie–Hellman [39] and ElGamal [45].

When groups are non-commutative, key exchange and public-key encryption seem much harder to achieve. To devise a public-key encryption scheme based on non-commutative group actions was proposed in [58] as an open problem. In a recent breakthrough [55], Hhan, Morimae and Yamakawa proposed the first public-key encryption scheme based on cryptographic non-commutative group actions, albeit requiring the ciphertexts to be quantum. Regarding key exchange, to the best of our knowledge, there were no proposals in vein of the Diffie–Hellman key exchange protocols before this work. Indeed, as discussed above, commutativity is used crucially in the original Diffie–Hellman protocol to ensure the key agreement.

While commutative groups admit more cryptographic functionalities more easily, there are several reasons to pursue non-commutative group actions, particularly in post-quantum cryptography. In the context of quantum algorithms, algorithmic problems underlying commutative group actions can usually be formulated as instances of the abelian hidden shift problem [33]. This allows for adapting the Kuperberg’s dihedral hidden subgroup algorithm [61] to obtain quantum subexponential-time algorithms such as for constructing elliptic curve isogenies [32] and some recent attacks [73, 23] on CSIDH [31].

On the other hand, the non-abelian group actions used in cryptography, such as linear code equivalence [18] and tensor isomorphism [58, 34, 81], can be cast as instances of the hidden subgroup problem with the ambient groups being symmetric and general linear groups. For such hidden subgroup problems, there is strong negative evidence [52, 68, 40, 41] for using the standard techniques from Shor’s algorithms [78] and Kuperberg’s sieve techniques [61] to obtain a polynomial-time or even subexponential-time quantum algorithm. These are referred to as the “strongest such insights we have about the limits of quantum algorithms” by Moore, Russell, and Vazirani [69].

Besides these considerations from quantum algorithms, the group action computation for CSIDH is relatively slow, and to speed it up requires considerable research (see e.g. [17, 56]). On the other hand, those non-abelian group actions based on lattices, tensors, and linear codes are fast to compute.

Quantum cryptography based on group actions. More advanced functionalities based on non-abelian group actions can be achieved in the context of quantum cryptography. We already mentioned the public-key encryption scheme with quantum ciphertexts by Hhan, Morimae and Yamakawa [55]. Another exciting recent development is the quantum money scheme based on non-abelian group action by Bostanci, Nehoran, and Zhandry [24].

1.2 Key exchange from commutativity to group laws: the framework

Our goal in this paper is to propose one approach to extend Diffie–Hellman key exchange protocols to beyond commutativity. The key notion here is the so-called group laws.

Group laws. A *law in a group* G is an equation that is satisfied by any assignments of variables by group elements in G . For example, $xy = yx$ is a law in abelian groups, or put in another way, abelian groups are groups satisfying the commutative law $xy = yx$. Therefore, on the one hand, laws can be used to define group classes, such as abelian groups, metabelian

groups, and solvable groups. On the other hand, there was considerable recent research on short laws for almost simple groups (such as symmetric groups) and general finite groups [60, 82, 25], and these works serve as a guidance for several considerations in this paper.

With the commutative law in mind, we reformulate Diffie–Hellman key exchange for abelian group actions as follows. Suppose an abelian group G acts on a set S from the right. For $g \in G$ and $s \in S$, we use $s * g$ to denote the result of $g \in G$ acting on $s \in S$. Following Diffie–Hellman, Alice randomly samples $a \in G$ and Bob samples $b \in G$. Alice then computes $s * a$ and sends it to Bob, while Bob computes $s * b$ and sends it to Alice. Alice and Bob computes $(s * b) * a$ and $(s * a) * b$ respectively and they reach a common key $(s * b) * a = s * (ba) = s * (ab) = (s * a) * b$, by group action axioms and the commutativity law. This formulation was observed independently by Couveignes [35] and Rostovtsev–Stolbunov [76, 80] in the context of isogeny-based cryptography.

Key exchange for general group actions. The Diffie–Hellman key exchange protocol can be readily generalized to actions of groups with laws. Indeed, let G be a group satisfying a law, that is an equation, of the form $u(a, b, c, \dots) = v(a, b, c, \dots)$, where u and v are words in variables a, b, c, \dots and their inverses. Then by assigning these variables to Alice and Bob appropriately, they can sample group elements from G , communicate according to u to get Alice’s key, and communicate according to v to get Bob’s key. The agreement of their keys is then ensured by the group law.

In Section 3, we formally present the above key exchange protocol for actions of groups with laws, and prove its security [10, 49], based on a *pseudorandom group action* assumption. One simplification we make there is to focus on laws involving just two variables, as a law in k variables can be transformed into another law in 2 variables of length polynomially bounded by the original law [25].

A natural question is then whether general groups admit some laws. That is, whether certain groups are “outlaws.” To start with, note that finite groups always admit some laws. For example, for a finite group G of order N , a naive law is $x^N = \text{id}$. Shorter laws with two variables exist, as Bradford and Thom showed the existence of laws in two variables of length $\tilde{O}(N^{2/3})$ for groups of order N [25]. Still, these laws would not be useful for our purpose, as the group order N is not polynomially bounded.

Key exchange for metabelian group actions. We now present the key exchange protocol for actions by metabelian groups. We believe that it is a nice example to illustrate the above ideas concretely. However, to the best of our knowledge, we do not have candidate cryptographic actions by metabelian groups; therefore, we will move to an instantiation based on symmetric group actions in the next subsection. We leave the search for cryptographic metabelian group actions as an intriguing open problem.

Denoting by $[a, b]$ the commutator of a, b . Recall that a group G is *metabelian*, if for any $a, b, c, d \in G$, $[a, b][c, d] = [c, d][a, b]$, i.e., $aba^{-1}b^{-1}cdc^{-1}d^{-1} = cdc^{-1}d^{-1}aba^{-1}b^{-1}$. In other words, the *metabelian* law is $xyx^{-1}y^{-1}uvu^{-1}v^{-1} = uvu^{-1}v^{-1}xyx^{-1}y^{-1}$ in variables x, y, u, v .

Note that $[a, b][c, d] = [c, d][a, b]$ is also equivalent to $[[a, b], [c, d]] = \text{id}$, which means metabelian groups are solvable groups of derived length ≤ 2 .¹ For example, dihedral groups and affine maps $x \rightarrow ax + b$ over a field are metabelian.

¹ Briefly speaking, the derived length of a group refers to the number of steps needed to reach the trivial group $\{\text{id}\}$ when repeatedly taking commutator subgroups.

Suppose a metabelian group G acts on a set S , with $g \in G$ sending $s \in S$ to $s * g \in S$. While the metabelian law can be used for Alice and Bob to reach a key agreement (see below), we need to be cautious about assigning a, b, c, d to Alice and Bob appropriately. Indeed, some assignments of $a, b, c, d \in G$ to Alice and Bob would get us back to the commutative setting, such as a and b to Alice, and c and d to Bob. Furthermore, we need to make sure that one sequence of communications ends with Alice holding the last group element, and the other sequence of communications ends with Bob holding the last group element, so they can apply these last elements to get their share of the key.

With these considerations in mind, we arrive at the following protocol.

1. Fix $s \in S$ as the public key.
2. Alice randomly samples $a, d \in G$, and Bob randomly samples $b, c \in G$, as their private keys.
3. Alice initiates $s * a$, and communicates with Bob in the following three rounds:
 - a. Alice $\xrightarrow{s*a}$ Bob $\xrightarrow{(s*a)*b}$ Alice. Let $t_1 = s * (ab)$.
 - b. Alice $\xrightarrow{t_1*a^{-1}}$ Bob $\xrightarrow{(t_1*a^{-1})*(b^{-1}c)}$ Alice. Let $t_2 = t_1 * (a^{-1}b^{-1}c)$.
 - c. Alice $\xrightarrow{t_2*d}$ Bob $\xrightarrow{(t_2*d)*c^{-1}}$ Alice. Let $t_3 = t_2 * (dc^{-1})$.
Alice then computes $t_4 = t_3 * d^{-1}$ as her secret key.
4. Bob initiates $s * c$, and communicates with Alice in the following three rounds.
 - a. Bob $\xrightarrow{s*c}$ Alice $\xrightarrow{(s*c)*d}$ Bob. Let $r_1 = s * (cd)$.
 - b. Bob $\xrightarrow{r_1*c^{-1}}$ Alice $\xrightarrow{(r_1*c^{-1})*d^{-1}a}$ Bob. Let $r_2 = r_1 * (c^{-1}d^{-1}a)$.
 - c. Bob $\xrightarrow{r_2*b}$ Alice $\xrightarrow{(r_2*b)*a^{-1}}$ Bob. Let $r_3 = r_2 * (ba^{-1})$.
Bob then computes $r_4 = r_3 * b^{-1}$ as his secret key.

Note that by assigning a, d to Alice and b, c to Bob, elements of the form $s * ([a, b])$ or $s * ([c, d])$ are never revealed during the execution of the protocol. This ensures that we do not get back to the commutative setting.

It is clear that Alice and Bob reach a key agreement: in Step 3, Alice and Bob follow the sequence on the left-hand side of the metabelian law, while in Step 4, Alice and Bob follow the sequence on the right-hand side of the metabelian law. Therefore, t_4 and r_4 , namely the secret keys of Alice and Bob, are the same.

The downside is that Alice and Bob need 6 rounds of communications. Alice performs 7 group actions, and so does Bob. In general, the group action is usually assumed to be efficiently computable. Recall that in the original Diffie–Hellman protocol, Alice and Bob only need 1 round of communication, and each of them performs 2 group actions. Still, when our interest is more on exploring the theoretical possibility of making use of non-commutative group actions, we could even accommodate polynomially many rounds of communications.

A more serious problem is that we don't know of concrete cryptographic metabelian group actions. That is, a group action suitable for cryptographic uses needs to demonstrate evidence for satisfying one-way [26] or pseudorandom [58] assumptions. At present, candidate cryptographic group actions are either abelian from isogeny based cryptography [31, 17], or highly non-abelian (almost simple) groups such as symmetric or general linear groups [58, 81, 34, 18]. Indeed, it would be of great interest to identify candidate cryptographic actions by non-abelian solvable groups.

1.3 Key exchange from commutativity to group laws: a concrete proposal

Key exchange for symmetric group actions. When we turn to symmetric groups S_n , the group of permutations of $[n] = \{1, 2, \dots, n\}$, some recent nice progress in group theory comes to our aid. In [60], Kozma and Thom demonstrated laws for S_n in two variables of length $\exp(\tilde{O}(\log(n)^4))$. This result builds on several important works, including [64, 54]. Actually, if a well-known conjecture of Babai [6] holds, the law length can be brought further down to $n^{O(\log \log(n))}$.

While the laws in [60] are close to polynomial in length, they may still be a bit too complicated to use in cryptographic protocols. Looking into the proofs in [60], an elementary fact about S_n turns out to be crucial: $1/n$ fraction of permutations are full-cycles. This means that $(xy)^n = \text{id}$, or equivalently $(xy)^{\lceil n/2 \rceil} = (y^{-1}x^{-1})^{\lfloor n/2 \rfloor}$, is a law that holds with probability (at least) $1/n$ when x and y are randomly sampled from S_n .²

Such probabilistic laws have been studied in the literature under the name of “almost laws”, and almost laws in two variables for S_n of length $\tilde{O}(n^8)$ were presented in [83] (see [25]). Here, we shall keep using $(xy)^{\lceil n/2 \rceil} = (y^{-1}x^{-1})^{\lfloor n/2 \rfloor}$ due to its simplicity and efficiency (length n compared to length $\tilde{O}(n^8)$ as in [83]).

This leads to the following protocol for key exchange based on symmetric group actions. Let S_n act on a set S , with $g \in S_n$ sending $s \in S$ to $s * g$. Alice randomly samples $g \in S_n$ and Bob randomly samples $h \in S_n$. They agree on $s \in S$ which can be made public. For Bob’s key, Alice and Bob compute and communicate $s * g, s * gh, \dots, s * (gh)^{\lceil n/2 \rceil - 1}$ in turn. In the last round, Alice sends $s * ((gh)^{\lceil n/2 \rceil - 1}g)$ to Bob, and Bob computes $s * ((gh)^{\lceil n/2 \rceil})$ as his key. For Alice’s key, Bob and Alice compute and communicate $s * h^{-1}, s * (h^{-1}g^{-1}), \dots, s * (h^{-1}g^{-1})^{\lfloor n/2 \rfloor - 1}$ in turn. In the last round, Bob sends $s * (h^{-1}g^{-1})^{\lfloor n/2 \rfloor - 1}h^{-1}$ to Alice, and Alice computes $s * (h^{-1}g^{-1})^{\lfloor n/2 \rfloor}$ as her key.

In the case of $(gh)^n = \text{id}$, such as gh being a full-cycle permutation, $s * ((gh)^{\lceil n/2 \rceil}) = s * (h^{-1}g^{-1})^{\lfloor n/2 \rfloor}$, that is, the keys of Alice and Bob agree. While the probability for it to hold is $1/n$, it suffices for a theoretical demonstration. We then apply a blackbox transformation from [49] to obtain key confirmation between Alice and Bob, and if it does not satisfy the key confirmation check, we will repeat the key exchange protocol by choosing a new pair of g and h , until $(gh)^n = \text{id}$ (cf. Section 5).

Security models for key exchange. There has been a long history in designing and analyzing the security of key exchange protocols. There are basically three approaches for defining the security of key exchange protocols [38]: indistinguishability approach [10], simulation-based security paradigm [79, 9], and universal composability [30] or reactive simulatability [74].

For our protocol, we follow Fischlin et al.’s approach [49], which follows the previous work by Bellare and Rogaway [10] via the indistinguishability approach. The reason is that in our protocol, especially in the instantiation, we need to make sure that the shared keys obtained from two parties after the execution of the session are the same, and hence we need our key exchange protocol to have *key confirmation* property. Basically, following [49], one needs a key exchange protocol Π that provides *key secrecy* and *match security* properties, then Fischlin et al. [49] provide a generic transformation to obtain a key exchange preserving those two properties with an additional *key confirmation* property.

² Note that when x and y are assigned with uniformly sampled random permutations, their product xy is also a uniformly random permutation.

New security assumptions. Due to non-commutativity, our key exchange protocol requires more rounds, and hence the transcript will consist of many elements of the form $s, s * a, s * ab, \dots$, etc. Therefore, a new hardness assumption is needed for the key secrecy proof to go through following [49]. Using the protocol based on metabelian groups in Section 1.2 as an example, we require that it is hard to distinguish between two distributions $\{(s, s * a, s * ab, s * aba^{-1}, \dots, s * aba^{-1}b^{-1}cdc^{-1}, s * aba^{-1}b^{-1}cdc^{-1}d^{-1})\}$ and $\{(s, s * a, s * ab, s * aba^{-1}, \dots, s * aba^{-1}b^{-1}cdc^{-1}, s * e)\}$ for random a, b, c, d, e . Note that here the shared key is $K = s * aba^{-1}b^{-1}cdc^{-1}d^{-1}$ and a random value is $s * e$. Besides that, we also need to consider DLP-like (Assumption 1) and CDH-like (Assumption 2) assumptions to ensure that the shared key cannot be obtained by the adversary from the transcript of the protocol.

Review of the group action underlying linear code equivalence. To instantiate the above key exchange protocol, we need a concrete group action suitable for cryptographic uses.

Symmetric group actions are an important topic in combinatorics and theoretical computer science. For example, the celebrated graph isomorphism problem can be formulated as the orbit problem of S_n acting on sets of size-2 subsets of $[n]$ [65]. Unfortunately (for cryptographers), graph isomorphism turns out to be (almost) easy both in practice [66] and in theory [3]. By contrast, the *linear code equivalence* problem is generally regarded as a candidate for cryptographic group actions, which asks whether two linear spaces are the same up to permutation of, and scalar multiplications on, the coordinates. In the literature, this version of the problem is usually referred to as *monomial code equivalence*, to distinguish from *permutation code equivalence*, in which scalar multiplications are not considered.

Several works show that permutation code equivalence is easy in some cases [77, 8]. Therefore, monomial code equivalence is preferred for cryptographic uses, as the current best practical algorithms for monomial code equivalence are the ones exploiting small-weight vectors by Leon and Beullens [62, 13]. For algorithms with worst-case asymptotic analyses, some recent works on monomial code equivalence design faster but still exponential-time algorithms [12], by exploiting some ideas from Babai's algorithm on permutation code equivalence [5]. Based on monomial code equivalence, the digital signature scheme LESS [18] was developed and submitted to the call for post-quantum digital signatures of NIST [7].

Viewing monomial code equivalence as a symmetric group action. Directly using monomial code equivalence has two issues. First, it is a monomial group action instead of a symmetric group action. This could be fixed relatively easily by utilizing group laws for the monomial group. The second and more serious issue is the following. During our protocol, a secret key needs to be used several times. Recall that it is straightforward to cast monomial code equivalence as the monomial group $\text{Mon}(n, q)$ acting on the set of m -dimensional codes in \mathbb{F}_q^n . With this modeling, in cryptographic protocols, the secret key is the monomial matrix M , which is a product of a diagonal matrix D and a permutation matrix P (as in e.g. [18]). A recent surprising discovery is that the secret monomial matrix cannot be used several times, not even twice [36, 28, 29]. The secret key reusing is the main bottleneck for using monomial code equivalence in our key exchange protocols.

In [43], it was shown that a symmetric group action can be extracted from the monomial code equivalence problem. Furthermore, it was suggested there that the techniques in [36, 28, 29] for breaking multiple uses of the secret keys in the monomial group action setting do not carry over to this symmetric group setting, at least not directly. This makes such group action suitable for instantiating the key exchange protocol as above.

2 Preliminaries

Notations. For $n \in \mathbb{N}$, $[n] := \{1, \dots, n\}$. For a finite set S , $s \leftarrow S$ means that the element s is sampled uniformly and independently at random from S . We denote by $|S|$ the cardinality of S . An adversary is a probabilistic polynomial time (PPT) algorithm.

Given a prime power q , let \mathbb{F}_q be the field consisting of q elements. Denote by $\text{Mat}(m \times n, q)$ the linear space of $m \times n$ matrices over \mathbb{F}_q , and $\text{Mat}(n, q) := \text{Mat}(n \times n, q)$. Let $\text{Mat}_f(m \times n, q)$ be the set of matrices in $\text{Mat}(m \times n, q)$ of rank $\min\{m, n\}$. Let \mathbb{F}_q^n be the linear space of length- n row vectors over \mathbb{F}_q . We denote by $\text{GL}(n, q)$ the group of $n \times n$ invertible matrices over the field \mathbb{F}_q , and $\text{D}(n, q)$ the group of $n \times n$ invertible diagonal matrices over \mathbb{F}_q . The symmetric group on $[n]$ is denoted by S_n . By interpreting $\sigma \in S_n$ as a permutation matrix, we view S_n as a subgroup of $\text{GL}(n, q)$. A matrix in $\text{Mat}(n, q)$ is *monomial*, if it is a product of an invertible diagonal matrix and a permutation matrix. In other words, it is a matrix where each row and each column has exactly one non-zero entries. The group of monomial matrices is denoted by $\text{Mon}(n, q)$.

Group actions. Let G be a group and S a set. A left action of G on S is a map $\alpha : G \times S \rightarrow S$ satisfying the following properties: (i) $\alpha(\text{id}, s) = s$ for all $s \in X$ and the identity element $\text{id} \in G$; and (ii) $\alpha(g, \alpha(h, s)) = \alpha(gh, s)$ for all $g, h \in G$ and $s \in X$.

A right action of G on S can be defined similarly. For convenience, we may write $*$ for a right group action α , where $s * g = \alpha(g, s)$.

Given a group action α of G on S , the orbit of an element $s \in S$ is defined as $\text{Orb}(s) := \{\alpha(g, s) : g \in G\}$. Note that if the group action is transitive, then $\text{Orb}(s) = S$ for any $s \in S$. The stabilizer of s is defined by $\text{Stab}(s) := \{g \in G : \alpha(g, s) = s\}$ which is a subgroup of G . The Orbit-Stabilizer theorem says that for a finite group G , $|G| = |\text{Stab}(s)| \cdot |\text{Orb}(s)|$.

In this paper, we shall mostly consider finite groups acting on finite sets. To use group actions in algorithms, we assume that group and set elements have natural encodings, as well as group operations, group actions, and random samplings of group and set elements can be efficiently computed; see [26, 58, 2] for more details and certain variations.

Group actions in cryptography. Following [26], we say that a group action is *one-way*, if for a random s , the function $f_s : G \rightarrow S$ defined by $f_s(g) := \alpha(g, s)$ is one-way.³ The one-way assumption is closely related to the following algorithmic problem, known as the orbit problem, or the vectorization problem, or the Group Action Inverse Problem (GAIP).

► **Definition 1 (GAIP).** *Given a group action $\alpha : G \times S \rightarrow S$, uniformly random $s \in S$, and uniformly random $t \in \text{Orb}(s)$, find $g \in G$ such that $\alpha(g, s) = t$.*

Another problem is called the *parallelization* problem, or the *Group Action Computational Diffie–Hellman (GACDH)* problem [67], defined as follows.

► **Definition 2 (GACDH).** *Given a group action $\alpha : G \times S \rightarrow S$, $s \in S$, $\alpha(g, s)$ and $\alpha(h, s)$ for uniformly-random $g, h \in G$, compute $\alpha(gh, s)$.*

A third problem, known as the Group Action Decisional Diffie–Hellman (GADDH) or pseudorandom group actions [58, 2], is defined as follows.

³ In [26], the definition of one-way group actions is slightly different, in that the function f_s only needs to be one-way for a chosen s .

► **Definition 3** (GADDH). *A group action $\alpha : G \times S \rightarrow S$ is pseudorandom, if no PPT adversary can distinguish between the following distributions: (1) (s, t) for $s, t \leftarrow S$ and (2) (s, t) for $s \leftarrow S$ and $t \leftarrow \text{Orb}(s)$.*

Note that for efficiency improvements of many constructions based on group actions, we need to consider variants of the above problems for more instances; see [17, 19] for details.

Security model and security properties. We consider two-party protocols which are defined by an interactive program Π that parties execute locally where parties belong to either a set of clients or a set of servers. For a detailed setting of the security model that underpins our key exchange protocols, we refer readers to our full version [44], which follows the definitions in [49, Sections II]. We also briefly recall the security properties required for key exchange protocols. As seen in [27, 49], the first two traditional properties are *key secrecy*, which ensures that (fresh) keys function as random secrets known only to the legitimate participants, and *match security*, which captures the correctness and soundness of the protocol. The last property is *key confirmation*, first defined in [49, Section III], which guarantees participants have the same key after the execution of the protocol/session. In what follows, we will just describe our protocol Π and prove that Π provides two essential properties, i.e., match security and key secrecy. This is because [49, Theorem 5.1] shows that the key confirmation property can be obtained by invoking the blackbox transformation in [49, Figure 4]. Again, please see our full version [44] for further details.

3 Key exchange for actions by groups with laws

3.1 The key exchange protocol

Words and laws. We define words and laws in two variables. This is partly for convenience and also due to the fact that k -variable laws can be turned to 2-variable laws by a polynomial blow-up in length [25]. We note that k -variable laws could be more efficient in practice.

► **Definition 4.** *Let x and y be two non-commutative variables. A word w in x and y is a string in x, y, x^{-1} , and y^{-1} . For $a \in \mathbb{N}$, we shall write a sequence of a many x 's (resp. x^{-1} 's) as x^a (resp. x^{-a}). The length of w is the number of alternations between x and y .*

► **Definition 5.** *A law in x and y is of the form $u = v$, where u and v are words in x and y . A (key-exchange) useful law is of the form $u = v$ in x and y where u ends with x and v ends with y . Let G be a group. We say that G satisfies the law $u = v$, if for any assignment of x and y with g and h in G , the resulting group elements on the left and right sides are equal.*

Key exchange protocol. Let G be a group and S be a set. Suppose G satisfies a useful law $u = v$, where $u = y^{b_1}x^{a_1} \dots y^{b_k}x^{a_k}$ and $v = x^{c_1}y^{d_1} \dots x^{c_\ell}y^{d_\ell}$ in variables x and y . Let $\alpha : G \times S \rightarrow S$ be a group action. Recall that from Section 2, we assume that the following can be computed in polynomial time: group operations (multiplication and inverse), uniform sampling of G and S , and the group action function.

Consider the following key exchange protocol between Alice and Bob. To start with, they fix some $s_0 \in S$ as the public key. Then Alice randomly samples $g \leftarrow G$, and Bob randomly samples $h \leftarrow G$. Now Alice does the following to obtain her secret key. Recall that by our assumption, we have a word $u = y^{b_1}x^{a_1} \dots y^{b_k}x^{a_k}$, where $a_i, b_i \in \mathbb{Z}$, and all but possibly b_1 are non-zero. If $b_1 \neq 0$, then $k \geq 1$. If $b_1 = 0$, then $k \geq 2$.

1. Bob computes $t_1 := \alpha(h^{b_1}, s_0)$ and sends it to Alice.
2. For $i \in [k - 1]$, do the following:
 - a. Alice computes $s_i := \alpha(g^{a_i}, t_i)$ and sends it to Bob.
 - b. Bob computes $t_{i+1} := \alpha(h^{b_{i+1}}, s_i)$ and sends it to Alice.
3. Alice computes $s_k = \alpha(g^{a_k}, t_k)$ as her secret key.

Bob can then obtain his secret key in the same way using v . This is just to note that v is of the form $x^{c_1}y^{d_1} \dots x^{c_\ell}y^{d_\ell}$, where $c_i, d_i \in \mathbb{Z}$, and all but possibly c_1 are non-zero. Specifically, Bob does the following.

1. Alice computes $s'_1 := \alpha(g^{c_1}, s_0)$ and sends it to Bob.
2. For $i \in [l - 1]$, do the following:
 - a. Bob computes $t'_i := \alpha(h^{d_i}, s'_i)$ and sends it to Alice.
 - b. Alice computes $s'_{i+1} := \alpha(g^{c_{i+1}}, t'_i)$ and sends it to Bob.
3. Bob computes $t'_l := \alpha(h^{d_l}, s'_l)$ as his secret key.

Key agreement and communication round number. Upon the generic transformation in [49, Figure 4], Alice’s key agrees with Bob’s. This is because Alice holds $\alpha(h^{b_1}g^{a_1} \dots h^{b_k}g^{a_k}, s_0)$ and Bob holds $\alpha(g^{c_1}h^{d_1} \dots g^{c_\ell}h^{d_\ell}, s_0)$. As $u = v$ is a law in G , we have that $h^{b_1}g^{a_1} \dots h^{b_k}g^{a_k} = g^{c_1}h^{d_1} \dots g^{c_\ell}h^{d_\ell}$, which ensures the key agreement. The number of communications between Alice and Bob depends on the lengths of u and v . In the above illustration, it is $2(k + \ell) - 2$.

3.2 Security proof

In this section, we provide the security proof of our key exchange protocol. For convenience, we first consider the first half transcript, as the second half transcript would follow the same reasoning. From the adversary Eve’s viewpoint, she knows s_0 (the public information), $t_1 = \alpha(h^{b_1}, s_0)$, $s_1 = \alpha(g^{a_1}, t_1)$, $t_2 = \alpha(h^{b_2}, s_1)$, \dots , $t_k = \alpha(h^{b_k}, s_{k-1})$. To recover Alice’s secret, she needs to be able to compute $s_k = \alpha(g^{a_k}, t_k)$. Note that s_0, \dots, s_{k-1} , t_1, \dots, t_k , and the exponents a_i and b_i , are known to Eve. So one approach for Eve to recover s_k is to compute g from the information $s_i = \alpha(g^{a_i}, t_i)$ for $i \in [k - 1]$. The hardness of this is ensured by a DLP-like assumption, which we call **Tuple Non-Commutative Discrete Logarithm Problem** or **TnDLP** for short, defined as the following.

► **Assumption 1 (TnDLP).** Let $\alpha : G \times S \rightarrow S$ be a group action. Let $s_0 \in S$, $P \subseteq \mathbb{Z}$, and $(b_1, a_1, \dots, b_k, a_k) \in P^{2k}$, and $(c_1, d_1, \dots, c_\ell, d_\ell) \in P^{2\ell}$. For uniformly-random elements $g, h \in G$, no computationally bounded adversaries can compute g or h with non-negligible probability, given the following information:

1. $t_1 = \alpha(h^{b_1}, s_0)$, $s_1 = \alpha(g^{a_1}, t_1)$, $t_2 = \alpha(h^{b_2}, s_1)$, \dots , $t_k = \alpha(h^{b_k}, s_{k-1})$, and
2. $s'_1 = \alpha(g^{c_1}, s_0)$, $t'_1 = \alpha(h^{d_1}, s'_1)$, $s'_2 = \alpha(g^{c_2}, t'_1)$, \dots , $s'_\ell = \alpha(g^{c_\ell}, t'_{\ell-1})$.

Another approach is to compute the shared key $s_k = \alpha(g^{a_k}, t_k)$ given the $s_i = \alpha(g^{a_i}, t_i)$ and for $i \in [k - 1]$ and $t_k = \alpha(h^{b_k}, s_{k-1})$ obtained from the transcript. This security is ensured by the hardness of the following CDH-like assumption, whose special case is called *weak unpredictable group action assumption* in [2]. We call this assumption to be **Tuple Non-Commutative Computational Diffie-Hellman** or **TnCDH** for short, defined as the following.

► **Assumption 2 (TnCDH).** Let $\alpha : G \times S \rightarrow S$ be a group action. Let $s_0 \in S$, $P \subseteq \mathbb{Z}$, and $(b_1, a_1, \dots, b_k, a_k) \in P^{2k}$, and $(c_1, d_1, \dots, c_\ell, d_\ell) \in P^{2\ell}$. For uniformly-random elements $g, h \in G$, no computationally bounded adversaries can compute $s_k := \alpha(h^{a_k}, t_k)$ or $t'_\ell := \alpha(h^{d_\ell}, s'_\ell)$ with non-negligible probability, given the following information:

1. $t_1 = \alpha(h^{b_1}, s_0)$, $s_1 = \alpha(g^{a_1}, t_1)$, $t_2 = \alpha(h^{b_2}, s_1)$, \dots , $t_k = \alpha(h^{b_k}, s_{k-1})$, and
2. $s'_1 = \alpha(g^{c_1}, s_0)$, $t'_1 = \alpha(h^{d_1}, s'_1)$, $s'_2 = \alpha(g^{c_2}, t'_1)$, \dots , $s'_\ell = \alpha(g^{c_\ell}, t'_{\ell-1})$.

Two assumptions TnDLP and TnCDH ensure that the shared key cannot be computed from the transcript in our key exchange protocol. Still, for the key secrecy property for our key exchange protocol, we need a DDH-like assumption, captured from the idea of the GACDH assumption, which we call Tuple Non-Commutative Decisional Diffie–Hellman Assumption or TnDDH for short, defined as the following.

► **Assumption 3 (TnDDH).** Given a group action $\alpha : G \times S \rightarrow S$, a fixed element $s_0 \in S$ and a finite set P of integer numbers. For $g, h \leftarrow_{\$} G$, the probability distributions between

- $T_0 = \{(s_0, \alpha(h^{b_1}, s_0), \alpha(h^{b_1}g^{a_1}, s_0), \dots, \alpha(h^{b_1}g^{a_1} \dots g^{a_{k-1}}h^{b_k}, s_0), \alpha(h^{b_1}g^{a_1} \dots h^{b_k}g^{a_k}, s_0))\}$ for $(b_1, a_1, \dots, b_k, a_k) \leftarrow_{\$} P^{2k}$; and
- $T_1 = \{(s_0, \alpha(h^{b_1}, s_0), \alpha(h^{b_1}g^{a_1}, s_0), \dots, \alpha(h^{b_1}g^{a_1} \dots g^{a_{k-1}}h^{b_k}, s_0), z)\}$ for $(b_1, a_1, \dots, b_k) \leftarrow_{\$} P^{2k-1}$ and $z \leftarrow_{\$} \text{Orb}(s_0)$

are computationally indistinguishable. Similarly, the probability distributions between

- $T'_0 = \{(s_0, \alpha(g^{c_1}, s_0), \alpha(g^{c_1}h^{d_1}, s_0), \dots, \alpha(g^{c_1}h^{d_1} \dots h^{d_{\ell-1}}g^{c_\ell}, s_0), \alpha(g^{c_1}h^{d_1} \dots g^{c_\ell}h^{d_\ell}, s_0))\}$ for $(c_1, d_1, \dots, c_\ell, d_\ell) \in P^{2\ell}$; and
- $T'_1 = \{(s_0, \alpha(g^{c_1}, s_0), \alpha(g^{c_1}h^{d_1}, s_0), \dots, \alpha(g^{c_1}h^{d_1} \dots h^{d_{\ell-1}}g^{c_\ell}, s_0), z)\}$ for $(c_1, d_1, \dots, c_\ell) \in P^{2\ell-1}$ and $z \leftarrow_{\$} \text{Orb}(s_0)$

are computationally indistinguishable.

Note that TnDDH assumption is in fact a tuple variant of the pseudorandom group action assumption GADDH defined in Section 2. Now we conclude the following theorem, whose full proof is presented in our full version [44].

► **Theorem 6.** *Given a group action $\alpha : G \times S \rightarrow S$ that satisfies the TnDDH assumption (Assumption 3), our key exchange protocol described in Section 3.1 has key secrecy and match security.*

► **Remark 7.** With essentially the same proof, Theorem 6 also holds under a variant of Assumption 3 in which g and h are chosen to satisfy a specific law in group G , e.g., $(gh)^n = \text{id}$. This will apply to our instantiation in Section 5.

4 Secret key reusing for the symmetric group action underlying monomial code equivalence

4.1 Monomial code equivalence as a symmetric group action

Monomial code equivalence. A linear code over \mathbb{F}_q is a subspace of \mathbb{F}_q^n . An m -dimensional code in \mathbb{F}_q^n can be represented by a generator matrix $C \in \text{Mat}_{\mathbb{F}}(m \times n, q)$ with rows spanning the code. The monomial code equivalence problem is formally stated as follows.

► **Problem 1 (Monomial code equivalence (MCE)).** For $n \in \mathbb{N}$, let $m \in [n]$. Let $C, C' \in \text{Mat}(m \times n, q)$ be of rank m . Decide if there exist $A \in \text{GL}(m, q)$ and $M \in \text{Mon}(n, q)$, such that $ACM = C'$. If yes, compute such A and M .

Since a monomial matrix is a product of an invertible diagonal matrix and a permutation matrix, it is equivalent to formulate monomial code equivalence as asking if there exist $A \in \text{GL}(m, q)$, $D \in D(n, q)$, and an $n \times n$ permutation matrix P , such that $ACDP = C'$.

A symmetric group action underlying monomial code equivalence. Following [43], we take a different perspective from above by formulating it as the symmetric group S_n acting on a set S consisting of equivalence classes. Specifically, for $C_1, C_2 \in \text{Mat}_f(m \times n, q)$, we define an equivalence relation \sim as $C_1 \sim C_2$ if and only if there exists some $A \in \text{GL}(m, q)$ and $D \in \text{D}(n, q)$ such that $C_1 = AC_2D$. This equivalence relation naturally partitions $\text{Mat}_f(m \times n, q)$ into equivalence classes of $\text{Mat}_f(m \times n, q)$ under the action of $\text{GL}(m, q) \times \text{D}(n, q)$. Note that this approach actually aligns with the natural viewpoint in that the monomial group $\text{Mon}(n, q)$ acts on the set of m -dimensional codes in \mathbb{F}_q^n , as the codes can be treated as the equivalence classes of $\text{Mat}_f(m \times n, q)$ under the action of $\text{GL}(m, q)$.

Denote by $[C]_\sim := \{ACD : A \in \text{GL}(m, q), D \in \text{D}(n, q)\}$ the equivalence class determined by \sim and corresponding to $C \in \text{Mat}_f(m \times n, q)$. Let $\text{Mat}_f(m \times n, q)/\sim = \{[C]_\sim : C \in \text{Mat}_f(m \times n, q)\}$ be the set of equivalence classes under \sim . Now we wish to define an action of S_n on $\text{Mat}_f(m \times n, q)/\sim$. For $P \in S_n$, since $[C]_\sim := \{ACD : A \in \text{GL}(m, q), D \in \text{D}(n, q)\}$ is a set of matrices, a natural map is to send $[C]_\sim$ to $[C]_\sim P := \{ACDP : A \in \text{GL}(m, q), D \in \text{D}(n, q)\}$. For S_n to act on $\text{Mat}_f(m \times n, q)/\sim$, we need to show that $[C]_\sim P$ is an element in $\text{Mat}_f(m \times n, q)/\sim$. This is ensured by the following proposition in [43].

► **Proposition 8** ([43, Proposition 1]). *Let $[C]_\sim \in \text{Mat}_f(m \times n, q)/\sim$, $P \in S_n$, and $[C]_\sim P$ be as above. Then $[C]_\sim P = [CP]_\sim$.*

Proposition 8 demonstrates that the map $\alpha : S_n \times \text{Mat}_f(m \times n, q)/\sim \rightarrow \text{Mat}_f(m \times n, q)/\sim$ by $P \in S_n$ sending $[C]_\sim$ to $[C]_\sim P = [CP]_\sim$ is a well-defined group action.

Canonical form algorithm. At the end of our key exchange protocol, Alice and Bob each get C_1 and C_2 in the same equivalence class $[C]_\sim \in \text{Mat}_f(m \times n, q)/\sim$. That is, there exist $A \in \text{GL}(m, q)$ and $D \in \text{D}(n, q)$ such that $C_1 = AC_2D$. To obtain a common key, Alice and Bob need to perform a *canonical form* algorithm to reach the same $C^* \in \text{Mat}_f(m \times n, q)/\sim$. Canonical forms have been extensively explored for graphs and matrix tuples [4, 75]. Let G be a group acting on a set S . A canonical form algorithm takes an input $s \in S$ and returns a canonical representative $s^* \in \text{Orb}(s) := \{g * s : \forall g \in G\}$. The term “canonical” means that for any input $s' \in \text{Orb}(s)$, the output of the canonical form algorithm must be the same element $s^* \in \text{Orb}(s)$. In our setting, we take $G = \text{GL}(m, q) \times \text{D}(n, q)$ and $S = \text{Mat}_f(m \times n, q)$. This particular problem has been well studied in [37, 43]. We refer readers to the following result in [43], which gives an algorithmic proof and strengthens [37, Proposition 4] by eliminating the probability of failure.

► **Proposition 9** ([43, Proposition 3]). *There is an $O(m^2n)$ -time canonical form algorithm for the action of $\text{GL}(m, q) \times \text{D}(n, q)$ on $\text{Mat}(m \times n, q)$.*

4.2 Reusing secret keys

We first recall the following key reusing problem in the monomial group action setting, which has been studied in [36, 28, 29].

► **Problem 2** (*t*-samples monomial code equivalence). For $n \in \mathbb{N}$, let $m \in [n]$. Let $\{C_i, C'_i : i \in [t]\} \subseteq \text{Mat}(m \times n, q)$. Decide if there exist $A_i \in \text{GL}(m, q)$ and $M \in \text{Mon}(n, q)$, such that $A_i C_i M = C'_i$ for all $i \in [t]$. If yes, compute such a common monomial matrix M .

For our key exchange protocol in Section 5, reusing secret keys for symmetric group actions gives rise to the following problem.

► **Problem 3** (General diagonal-masked linear code equivalence (DmLCE)). For $n \in \mathbb{N}$, let $m \in [n]$. Let $\{C_i : i \in [n]\} \subseteq \text{Mat}_f(m \times n, q)$. Decide if there exist $\{A_i : i \in [n-1]\} \subseteq \text{GL}(m, q)$, $\{D_i : i \in [n-1]\} \subseteq \text{D}(n, q)$, and an $n \times n$ permutation matrix P , such that $A_i C_i D_i P = C_{i+1}$ for all $i \in [n-1]$. If yes, compute such a common permutation matrix P .

We note that Problem 2 reuses the same monomial group action from $\text{Mon}(n, q)$ and changes only the general linear group action from $\text{GL}(m, q)$ in each round. By contrast, Problem 3 reuses the same symmetric group action from S_n , but keeps changing the group action from a composite group $\text{GL}(m, q) \times \text{D}(n, q)$ to strengthen the hiding of secret information in each round. For further discussion on the comparison of these two problems, and on why the approaches proposed in [28, 29] for attacking Problem 2 cannot be effectively applied to Problem 3, we refer readers to our full version [44], as well as [43] whose underlying security problems are analogous to Problem 3. In our full version [44], we also present concrete experiments for solving Problem 3 by Gröbner basis methods to support its hardness.

5 An instantiation based on monomial code equivalence

In this section, we present a concrete key exchange protocol following the framework in Section 3 and based on the symmetric group action underlying monomial code equivalence in Section 4, using a law of the form $(PQ)^{\lceil n/2 \rceil} = (Q^{-1}P^{-1})^{\lfloor n/2 \rfloor}$ for $P, Q \in S_n$.

To prepare for our key exchange protocol based on this group action, we recall the following from symmetric groups. Let S_n be the symmetric group on $[n]$. For distinct $i_1, \dots, i_k \in [n]$, denote by (i_1, \dots, i_k) the permutation sending i_1 to i_2, \dots, i_{k-1} to i_k , and i_k to i_1 , and leaving others fixed. Recall that every permutation admits a cycle decomposition. A permutation is called a *full-cycle*, if $P = (1, i_1, \dots, i_{n-1})$ where $i_1, \dots, i_{n-1} \neq 1$ are distinct. Note that a full-cycle permutation is of order n , i.e., $P^n = \text{id}$. If the product PQ of two permutations P and Q is a full-cycle, then by $(PQ)^n = \text{id}$, we have a law of the form $(PQ)^{\lceil n/2 \rceil} = (Q^{-1}P^{-1})^{\lfloor n/2 \rfloor}$. In particular, if n is prime, then P being a full-cycle is equivalent to $P^n = \text{id}$.

► **Fact 1.** *Let P and Q be uniformly randomly sampled permutations from S_n . The probability that PQ forms a full-cycle is $1/n$.*

We can then describe the key exchange protocol based on this symmetric group action in Algorithm 1. Since we are only interested in the case when PQ forms a full-cycle such that $(PQ)^n = \text{id}$ where P, Q are uniformly randomly sampled permutations from S_n , to avoid the cases when PQ 's order is a divisor of n , we simply choose n to be prime. Though, for composite n , it actually just takes a low risk according to [71, Theorem 1.1]: for sufficiently large n and a uniformly randomly sampled permutation $P \in S_n$, the probability that $P^n = \text{id}$ is $\frac{1}{n} + \frac{c}{n^2} + O\left(\frac{1}{n^{2.5-o(1)}}\right)$ where $c = 0$ if n is odd and $c = 2$ if n is even. This shows that, given $P^n = \text{id}$, the probability that P has order dividing n is much smaller than the probability that P has order exactly n .

► **Remark 10.** The use of full-cycle permutations raises the question of whether the monomial code equivalence problem would be easier if the underlying permutation is a full-cycle. However, this is not the case, because we can reduce from general permutations to full-cycles by a random reduction: if C_1 and C_2 are related by a permutation Q , we can apply a random permutation R to C_2 to get C_3 , so with probability $1/n$, C_1 and C_3 are related by a full-cycle.

Algorithm 1 Key exchange procedure.

Input:

Public: The action $\alpha : S_n \times \text{Mat}_f(m \times n, q)/\sim \rightarrow \text{Mat}_f(m \times n, q)/\sim$, and $C_{\text{public}} \in \text{Mat}_f(m \times n, q)$, with $m \leq n$ and n is an odd prime.

Output: Key pair $C_n, C'_n \in \text{Mat}_f(m \times n, q)$ satisfying $C_n \sim C'_n$.

- 1 Alice randomly samples $P \leftarrow \$ S_n$, $A_0 \in \text{GL}(m, q)$ and $D_0 \in \text{D}(n, q)$, then she computes $C_1 \leftarrow A_0 C_{\text{public}} D_0 P$ and sends C_1 to Bob.
 - 2 Bob randomly samples $Q \leftarrow \$ S_n$.
 - 3 **for** $i \in [\frac{n+1}{2} - 1]$ **do**
 - 4 Bob randomly samples $A_{2i-1} \in \text{GL}(m, q)$ and $D_{2i-1} \in \text{D}(n, q)$, then he computes $C_{2i} \leftarrow A_{2i-1} C_{2i-1} D_{2i-1} Q$ and sends C_{2i} to Alice.
 - 5 Alice randomly samples $A_{2i} \in \text{GL}(m, q)$ and $D_{2i} \in \text{D}(n, q)$, then she computes $C_{2i+1} \leftarrow A_{2i} C_{2i} D_{2i} P$ and sends C_{2i+1} to Bob.
 - 6 **end**
 - 7 Bob randomly samples $A_n \in \text{GL}(m, q)$ and $D_n \in \text{D}(n, q)$, then he computes $C_{n+1} \leftarrow A_n C_n D_n Q$ and computes the canonical form of C_{n+1} as in Proposition 9. **This settles one key in the key pair.**
 - 8 Bob randomly samples $A'_0 \in \text{GL}(m, q)$ and $D'_0 \in \text{D}(n, q)$, then he computes $C'_1 \leftarrow A'_0 C_{\text{public}} D'_0 Q^{-1}$ and sends C'_1 to Alice.
 - 9 **for** $i \in [\frac{n-1}{2} - 1]$ **do**
 - 10 Alice randomly samples $A'_{2i-1} \in \text{GL}(m, q)$ and $D'_{2i-1} \in \text{D}(n, q)$, then she computes $C'_{2i} \leftarrow A'_{2i-1} C'_{2i-1} D'_{2i-1} P^{-1}$ and sends C'_{2i} to Bob.
 - 11 Bob randomly samples $A'_{2i} \in \text{GL}(m, q)$ and $D'_{2i} \in \text{D}(n, q)$, then he computes $C'_{2i+1} \leftarrow A'_{2i} C'_{2i} D'_{2i} Q^{-1}$ and sends C'_{2i+1} to Alice.
 - 12 **end**
 - 13 Alice randomly samples $A'_{n-2} \in \text{GL}(m, q)$ and $D'_{n-2} \in \text{D}(n, q)$, then she computes $C'_{n-1} \leftarrow A'_{n-2} C'_{n-2} D'_{n-2} P^{-1}$ and computes the canonical form of C'_{n-1} as in Proposition 9. **This settles another key in the key pair.**
 - 14 Apply the blackbox transformation in [49, Figure 4] for the key to be the canonical form of C_{n+1} for Bob and the canonical form of C'_{n-1} for Alice. If it aborts, then return to Step 1.
-

► **Remark 11.** In Algorithm 1, we expect that Alice’s permutation $P \in S_n$ and Bob’s permutation $Q \in S_n$ satisfy that $(PQ)^n = \text{id}$. To obtain such P and Q , Alice and Bob each need to uniformly sample a random permutation from S_n (in Steps 1 and 2, respectively), and the probability of $(PQ)^n = \text{id}$ for prime n is $1/n$ by Fact 1. At Step 14, if $(PQ)^n \neq \text{id}$, then the key confirmation will not succeed, causing it to abort and return to Step 1 so that Alice and Bob can re-sample different P and Q . The algorithm will repeat until $(PQ)^n = \text{id}$.

We now establish the correctness of Algorithm 1, whose proof appears in our full version [44]. The security assumption for this instantiation can be obtained from Assumption 3 by setting $G = S_n$ and $S = \text{Mat}_f(m \times n, q)/\sim$ with $(gh)^n = \text{id}$ (see Remark 7). We also leave further cryptanalysis on its computational version, Problem 3, as an intriguing open problem.

► **Proposition 12** (Correctness of the key exchange protocol). *There exist $A \in \text{GL}(m, q)$ and $D \in \text{D}(n, q)$ such that $AC_{n+1}D = C'_{n-1}$.*

The communication round number. It is expected to have $(2n^2 - 2n)$ many communication rounds, since the probability of obtaining a full-cycle permutation is $1/n$ and hence we may need to repeat the key exchange n many times.

References

- 1 Michel Abdalla, Thorsten Eisenhofer, Eike Kiltz, Sabrina Kunzweiler, and Doreen Riepel. Password-authenticated key exchange from group actions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 699–728. Springer, 2022. doi:10.1007/978-3-031-15979-4_24.
- 2 Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 411–439. Springer, 2020. doi:10.1007/978-3-030-64834-3_14.
- 3 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. doi:10.1145/2897518.2897542.
- 4 László Babai. Canonical form for graphs in quasipolynomial time: preliminary report. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 1237–1246. ACM, 2019. doi:10.1145/3313276.3316356.
- 5 László Babai, Paolo Codenotti, Joshua A. Grochow, and Youming Qiao. Code equivalence and group isomorphism. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1395–1408, 2011. doi:10.1137/1.9781611973082.107.
- 6 László Babai and Ákos Seress. On the diameter of permutation groups. *European journal of combinatorics*, 13(4):231–243, 1992. doi:10.1016/S0195-6698(05)80029-0.
- 7 Marco Baldi, Alessandro Barengi, Luke Beckwith, Jean-François Biasse, Andre Esser, Kris Gaj, Kamyar Mohajerani, Gerardo Pelosi, Edoardo Persichetti, Markku-Juhani Saarinen, Paolo Santini, and Robert Wallace. LESS: Linear equivalence signature scheme, 2023. URL: <https://www.less-project.com/LESS-2023-08-18.pdf>.
- 8 Magali Bardet, Ayoub Otmani, and Mohamed Saeed-Taha. Permutation code equivalence is not harder than graph isomorphism when hulls are trivial. In *IEEE International Symposium on Information Theory, ISIT 2019, Paris, France, July 7-12, 2019*, pages 2464–2468. IEEE, 2019. doi:10.1109/ISIT.2019.8849855.
- 9 Mihir Bellare, Ran Canetti, and Hugo Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols (extended abstract). In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 419–428. ACM, 1998. doi:10.1145/276698.276854.
- 10 Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 232–249. Springer, 1993. doi:10.1007/3-540-48329-2_21.
- 11 Benjamin Benčína, Alessandro Budroni, Jesús-Javier Chi-Domínguez, and Mukul Kulkarni. Properties of lattice isomorphism as a cryptographic group action. In *International Conference on Post-Quantum Cryptography*, pages 170–201. Springer, 2024. doi:10.1007/978-3-031-62743-9_6.
- 12 Huck Bennett, Drisana Bhatia, Jean-François Biasse, Medha Durisheti, Lucas LaBuff, Vincenzo Pallozzi Lavorante, and Philip Waitkevich. Asymptotic improvements to provable algorithms for the code equivalence problem. *Cryptology ePrint Archive*, 2025.

- 13 Ward Beullens. Not enough LESS: an improved algorithm for solving code equivalence problems over \mathbb{F}_q . In Orr Dunkelman, Michael J. Jacobson Jr., and Colin O’Flynn, editors, *Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers*, volume 12804 of *Lecture Notes in Computer Science*, pages 387–403. Springer, 2020. doi:10.1007/978-3-030-81652-0_15.
- 14 Ward Beullens, Luca De Feo, Steven D Galbraith, and Christophe Petit. Proving knowledge of isogenies: a survey. *Designs, Codes and Cryptography*, 91(11):3425–3456, 2023. doi:10.1007/s10623-023-01243-3.
- 15 Ward Beullens, Samuel Dobson, Shuichi Katsumata, Yi-Fu Lai, and Federico Pintore. Group signatures and more from isogenies and lattices: Generic, simple, and efficient. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part II*, volume 13276 of *Lecture Notes in Computer Science*, pages 95–126. Springer, 2022. doi:10.1007/978-3-031-07085-3_4.
- 16 Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and falaf: Logarithmic (linkable) ring signatures from isogenies and lattices. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 464–492. Springer, 2020. doi:10.1007/978-3-030-64834-3_16.
- 17 Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In *Advances in Cryptology - ASIACRYPT 2019*, volume 11921 of *Lecture Notes in Computer Science*, pages 227–247. Springer, 2019. doi:10.1007/978-3-030-34578-5_9.
- 18 Jean-François Biasse, Giacomo Micheli, Edoardo Persichetti, and Paolo Santini. LESS is more: Code-based signatures without syndromes. In Abderrahmane Nitaj and Amr M. Youssef, editors, *Progress in Cryptology - AFRICACRYPT 2020 - 12th International Conference on Cryptology in Africa, Cairo, Egypt, July 20-22, 2020, Proceedings*, volume 12174 of *Lecture Notes in Computer Science*, pages 45–65. Springer, 2020. doi:10.1007/978-3-030-51938-4_3.
- 19 Markus Bläser, Zhili Chen, Dung Hoang Duong, Antoine Joux, Tuong Ngoc Nguyen, Thomas Plantard, Youming Qiao, Willy Susilo, and Gang Tang. On digital signatures based on group actions: QROM security and ring signatures. In Markku-Juhani O. Saarinen and Daniel Smith-Tone, editors, *Post-Quantum Cryptography - 15th International Workshop, PQCrypto 2024, Oxford, UK, June 12-14, 2024, Proceedings, Part I*, volume 14771 of *Lecture Notes in Computer Science*, pages 227–261. Springer, 2024. doi:10.1007/978-3-031-62743-9_8.
- 20 Markus Bläser, Dung Hoang Duong, Anand Kumar Narayanan, Thomas Plantard, Youming Qiao, Arnaud Sipasseuth, and Gang Tang. The ALTEQ Signature Scheme: Algorithm Specifications and Supporting Documentation. NIST Post-Quantum Additional Digital Signature Schemes Submission, 2023. Authors listed on the official NIST specification document.
- 21 Dan Boneh. The decision Diffie-Hellman problem. In *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, pages 48–63, 1998. doi:10.1007/BFb0054851.
- 22 Dan Boneh, Jiabin Guan, and Mark Zhandry. A lower bound on the length of signatures based on group actions and generic isogenies. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 507–531. Springer, 2023. doi:10.1007/978-3-031-30589-4_18.
- 23 Xavier Bonnetain and André Schrottenloher. Quantum security analysis of CSIDH. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 493–522. Springer, 2020. doi:10.1007/978-3-030-45724-2_17.

- 24 John Bostanci, Barak Nehoran, and Mark Zhandry. A general quantum duality for representations of groups with applications to quantum money, lightning, and fire. In *Proceedings of the 57th ACM Symposium on Theory of Computing (STOC 2025)*, pages 189–200, 2025.
- 25 Henry Bradford and Andreas Thom. Short laws for finite groups and residual finiteness growth. *Transactions of the American Mathematical Society*, 371(9):6447–6462, 2019. doi:10.1090/tran/7518.
- 26 Gilles Brassard and Moti Yung. One-way group actions. In *Advances in Cryptology - CRYPTO 1990*, pages 94–107, 1990. doi:10.1007/3-540-38424-3_7.
- 27 Christina Brzuska, Marc Fischlin, Bogdan Warinschi, and Stephen C Williams. Composability of bellare-rogaway key exchange protocols. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 51–62, 2011. doi:10.1145/2046707.2046716.
- 28 Alessandro Budroni, Jesús-Javier Chi-Domínguez, Giuseppe D’Alconzo, Antonio J. Di Scala, and Mukul Kulkarni. Don’t use it twice! solving relaxed linear equivalence problems. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology - ASIACRYPT 2024 - 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9-13, 2024, Proceedings, Part VIII*, volume 15491 of *Lecture Notes in Computer Science*, pages 35–65. Springer, 2024. doi:10.1007/978-981-96-0944-4_2.
- 29 Alessandro Budroni and Andrea Natale. On the sample complexity of linear code equivalence for all code rates. *Cryptography and Communications*, pages 1–23, 2025.
- 30 Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 136–145. IEEE Computer Society, 2001. doi:10.1109/SFCS.2001.959888.
- 31 Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)*, pages 395–427. Springer, 2018. doi:10.1007/978-3-030-03332-3_15.
- 32 Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014. doi:10.1515/jmc-2012-0016.
- 33 Andrew M. Childs and Wim van Dam. Quantum algorithms for algebraic problems. *Rev. Mod. Phys.*, 82:1–52, January 2010. doi:10.1103/RevModPhys.82.1.
- 34 Tung Chou, Ruben Niederhagen, Edoardo Persichetti, Tovoheri Hajatiana Randrianarisoa, Krijn Reijnders, Simona Samardjiska, and Monika Trimoska. Take your meds: Digital signatures from matrix code equivalence. In *Progress in Cryptology - AFRICACRYPT 2023*, 2023. doi:10.1007/978-3-031-37679-5_2.
- 35 Jean Marc Couveignes. Hard homogeneous spaces. *Cryptology ePrint Archive*, 2006. URL: <http://eprint.iacr.org/2006/291>.
- 36 Giuseppe D’Alconzo and Antonio J Di Scala. Representations of group actions and their applications in cryptography. *Finite Fields and Their Applications*, 99:102476, 2024. doi:10.1016/j.ffa.2024.102476.
- 37 Giuseppe D’Alconzo, Alessio Meneghetti, and Edoardo Signorini. Group factorisation for smaller signatures from cryptographic group actions. *Cryptology ePrint Archive*, 2024. URL: <https://eprint.iacr.org/2024/1510>.
- 38 Anupam Datta, Ante Derek, John C. Mitchell, and Bogdan Warinschi. Key exchange protocols: Security definition, proof method and applications. *Cryptology ePrint Archive*, 2006. URL: <https://eprint.iacr.org/2006/056>.
- 39 Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976. doi:10.1109/TIT.1976.1055638.
- 40 Hang Dinh, Cristopher Moore, and Alexander Russell. McEliece and Niederreiter cryptosystems that resist quantum Fourier sampling attacks. In *Advances in Cryptology - CRYPTO 2011*, pages 761–779, 2011. doi:10.1007/978-3-642-22792-9_43.

- 41 Hang T. Dinh, Christopher Moore, and Alexander Russell. Limitations of single coset states and quantum algorithms for code equivalence. *Quantum Information & Computation*, 15(3&4):260–294, 2015. doi:10.26421/QIC15.3-4-4.
- 42 Léo Ducas, Eamonn W. Postlethwaite, Ludo N. Pulles, and Wessel van Woerden. HAWK: Module lip makes lattice signatures fast, compact and simple. In *Advances in Cryptology – ASIACRYPT 2022*, volume 13794 of *Lecture Notes in Computer Science*, pages 65–94. Springer, 2022. doi:10.1007/978-3-031-22972-5_3.
- 43 Dung Hoang Duong, Xuan Thanh Khuc, Youming Qiao, Willy Susilo, and Chuanqi Zhang. Blind signatures from cryptographic group actions. *Cryptology ePrint Archive*, 2025. URL: <https://eprint.iacr.org/2025/397>.
- 44 Dung Hoang Duong, Youming Qiao, and Chuanqi Zhang. Diffie-Hellman key exchange from commutativity to group laws. *Cryptology ePrint Archive*, 2025. URL: <https://eprint.iacr.org/2025/1677>.
- 45 Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985. doi:10.1007/3-540-39568-7_2.
- 46 Luca De Feo and Steven D. Galbraith. Seasign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, volume 11478 of *Lecture Notes in Computer Science*, pages 759–789. Springer, 2019. doi:10.1007/978-3-030-17659-4_26.
- 47 Luca De Feo and Michael Meyer. Threshold schemes from isogeny assumptions. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part II*, volume 12111 of *Lecture Notes in Computer Science*, pages 187–212. Springer, 2020. doi:10.1007/978-3-030-45388-6_7.
- 48 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. doi:10.1007/3-540-47721-7_12.
- 49 Marc Fischlin, Felix Günther, Benedikt Schmidt, and Bogdan Warinschi. Key confirmation in key exchange: A formal treatment and implications for TLS 1.3. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*, pages 452–469. IEEE Computer Society, 2016. doi:10.1109/SP.2016.34.
- 50 Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991. doi:10.1145/116825.116852.
- 51 Joshua A. Grochow and Youming Qiao. Isomorphism problems for tensors, groups, and cubic forms: completeness and reductions, 2019. arXiv:1907.00309 [cs.CC]. doi:10.48550/arXiv.1907.00309.
- 52 Sean Hallgren, Christopher Moore, Martin Rötteler, Alexander Russell, and Pranab Sen. Limitations of quantum coset states for graph isomorphism. *J. ACM*, 57(6):34:1–34:33, November 2010. doi:10.1145/1857914.1857918.
- 53 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. doi:10.1137/1.9781611973402.29.
- 54 Harald A Helfgott and Ákos Seress. On the diameter of permutation groups. *Annals of mathematics*, pages 611–658, 2014. doi:10.4007/annals.2014.179.2.4.
- 55 Minki Hhan, Tomoyuki Morimae, and Takashi Yamakawa. From the hardness of detecting superpositions to cryptography: Quantum public key encryption and commitments. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 639–667. Springer, 2023. doi:10.1007/978-3-031-30545-0_22.

- 56 Marc Houben. Deterministic algorithms for class group actions. In *Advances in Cryptology – CRYPTO 2025*, volume 16000 of *Lecture Notes in Computer Science*, pages 100–130. Springer, 2025. doi:10.1007/978-3-032-01855-7_4.
- 57 Ren Ishibashi and Kazuki Yoneyama. Compact password authenticated key exchange from group actions. In Leonie Simpson and Mir Ali Rezazadeh Bae, editors, *Information Security and Privacy - 28th Australasian Conference, ACISP 2023, Brisbane, QLD, Australia, July 5-7, 2023, Proceedings*, volume 13915 of *Lecture Notes in Computer Science*, pages 220–247. Springer, 2023. doi:10.1007/978-3-031-35486-1_11.
- 58 Zhengfeng Ji, Youming Qiao, Fang Song, and Aaram Yun. General linear group action on tensors: A candidate for post-quantum cryptography. In *Theory of cryptography conference*, pages 251–281. Springer, 2019. doi:10.1007/978-3-030-36030-6_11.
- 59 Shuichi Katsumata, Yi-Fu Lai, Jason T. LeGrow, and Ling Qin. CSI -otter: Isogeny-based (partially) blind signatures from the class group action with a twist. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 729–761. Springer, 2023. doi:10.1007/978-3-031-38548-3_24.
- 60 Gady Kozma and Andreas Thom. Divisibility and laws in finite simple groups. *Mathematische Annalen*, 364(1):79–95, 2016. doi:10.1007/s00208-015-1201-4.
- 61 Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM J. Comput.*, 35(1):170–188, 2005. doi:10.1137/S0097539703436345.
- 62 Jeffrey S. Leon. Computing automorphism groups of error-correcting codes. *IEEE Trans. Information Theory*, 28(3):496–510, 1982. doi:10.1109/TIT.1982.1056498.
- 63 Antonin Leroux and Maxime Roméas. Updatable encryption from group actions. In Markku-Juhani O. Saarinen and Daniel Smith-Tone, editors, *Post-Quantum Cryptography - 15th International Workshop, PQCrypto 2024, Oxford, UK, June 12-14, 2024, Proceedings, Part II*, volume 14772 of *Lecture Notes in Computer Science*, pages 20–53. Springer, 2024. doi:10.1007/978-3-031-62746-0_2.
- 64 Martin W Liebeck. On minimal degrees and base sizes of primitive permutation groups. *Archiv der Mathematik*, 43(1):11–15, 1984. doi:10.1007/BF01193603.
- 65 Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.*, 25(1):42–65, 1982. doi:10.1016/0022-0000(82)90009-5.
- 66 Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *J. Symb. Comput.*, 60:94–112, 2014. doi:10.1016/j.jsc.2013.09.003.
- 67 Hart Montgomery and Mark Zhandry. Full quantum equivalence of group action dlog and cdh, and more. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part I*, volume 13791 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2022. doi:10.1007/978-3-031-22963-3_1.
- 68 Christopher Moore, Alexander Russell, and Piotr Sniady. On the impossibility of a quantum sieve algorithm for graph isomorphism. *SIAM J. Comput.*, 39(6):2377–2396, 2010. doi:10.1137/080724101.
- 69 Christopher Moore, Alexander Russell, and Umesh Vazirani. A classical one-way function to confound quantum adversaries. *arXiv preprint*, 2007. arXiv:quant-ph/0701115.
- 70 Alexei G Myasnikov, Vladimir Shpilrain, and Alexander Ushakov. *Non-commutative cryptography and complexity of group-theoretic problems*, volume 177 of *Mathematical Surveys and Monographs*. American Mathematical Soc., 2011. doi:10.1090/surv\%2F177.
- 71 Alice C Niemeyer and Cheryl E Praeger. On permutations of order dividing a given integer. *Journal of Algebraic Combinatorics*, 26:125–142, 2007. doi:10.1007/s10801-007-0056-5.

- 72 Jacques Patarin and Valérie Nachev. Commutativity, associativity, and public key cryptography. In Jerzy Kaczorowski, Josef Pieprzyk, and Jacek Pomykala, editors, *Number-Theoretic Methods in Cryptology - First International Conference, NuTMiC 2017, Warsaw, Poland, September 11-13, 2017, Revised Selected Papers*, volume 10737 of *Lecture Notes in Computer Science*, pages 104–117. Springer, 2017. doi:10.1007/978-3-319-76620-1_7.
- 73 Chris Peikert. He gives c-sieves on the CSIDH. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 463–492. Springer, 2020. doi:10.1007/978-3-030-45724-2_16.
- 74 Birgit Pfitzmann and Michael Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *2001 IEEE Symposium on Security and Privacy, Oakland, California, USA May 14-16, 2001*, pages 184–200. IEEE Computer Society, 2001. doi:10.1109/SECPRI.2001.924298.
- 75 Youming Qiao and Xiaorui Sun. Canonical forms for matrix tuples in polynomial time. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 780–789. IEEE, 2024. doi:10.1109/FOCS61266.2024.00054.
- 76 Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. *IACR Cryptol. ePrint Arch.*, page 145, 2006. URL: <http://eprint.iacr.org/2006/145>.
- 77 Nicolas Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Trans. Information Theory*, 46(4):1193–1203, 2000. doi:10.1109/18.850662.
- 78 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. doi:10.1109/SFCS.1994.365700.
- 79 Victor Shoup. On formal models for secure key exchange. *Cryptology ePrint Archive*, page 12, 1999. URL: <http://eprint.iacr.org/1999/012>.
- 80 Anton Stolbunov. Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. *Adv. Math. Commun.*, 4(2):215–235, 2010. doi:10.3934/AMC.2010.4.215.
- 81 Gang Tang, Dung Hoang Duong, Antoine Joux, Thomas Plantard, Youming Qiao, and Willy Susilo. Practical post-quantum signature schemes from isomorphism problems of trilinear forms. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III*, volume 13277 of *Lecture Notes in Computer Science*, pages 582–612. Springer, 2022. doi:10.1007/978-3-031-07082-2_21.
- 82 Andreas Thom. About the length of laws for finite groups. *Israel Journal of Mathematics*, 219:469–478, 2017. doi:10.1007/s11856-017-1487-x.
- 83 Christiane Zyrus. *Almost laws for finite simple groups*. PhD thesis, Technische Universität Dresden, 2020.