A Quasipolynomial Quantum Algorithm for the Vector Dihedral Coset Problem with Smooth Modulus

Aparna Gupte
MIT

November 17, 2026

Abstract

We give a quantum algorithm for the vector dihedral coset problem over $\mathbb{Z}_q^n$, in which the goal is to recover $\mathbf{s} \in \mathbb{Z}_q^n$ given independent samples of the form $\frac{1}{\sqrt{2}}\big(|0, \mathbf{x}\rangle + |1, \mathbf{x} + \mathbf{s}\rangle\big)$, for $\mathbf{x} \sim \mathbb{Z}_q^n$. When the modulus q is B-smooth, the algorithm uses $n^{O(B^{1+o(1)} \log q)} B^{O(B^{2+o(1)} \log q)}$ samples and time. For $B = \mathrm{polylog}(n)$ and $q = \mathrm{poly}(n)$, this gives a quasipolynomial time- and sample-complexity. This extends the previously known quasipolynomial-time result for power-of-two moduli (Bonnetain–Naya-Plasencia, ASIACRYPT 2018; Bai–Jangir–Kirshanova–Ngo–Youmans, CRYPTO 2025).

The algorithm proceeds by reducing the modulus one prime factor $p|q$ at a time. The key ingredient is a sieving procedure that clears the non-linear terms in the phase modulo p via an algorithm of Ivanyos and Santha (TCS 2017) that finds non-zero $\mathbf{u} \in \mathbb{F}_p^\ell$ such that $\sum_{i \in [\ell]} u_i^d \mathbf{a}_i = 0 \mod p$ for degrees $1 \le d \le p-1$, given vectors $\mathbf{a}_1, \ldots, \mathbf{a}_\ell \in \mathbb{F}_p^n$ as input.

The algorithm also applies to the extrapolated dihedral coset problem (EDCP) introduced by Brakerski–Kirshanova–Stehlé–Wen (PKC 2018), which is known to be equivalent to LWE in the polynomial sample regime. However, our result does not yield a new attack on LWE in any regime, precisely because the known reduction from LWE to EDCP provides only polynomially many clean samples, whereas our algorithm requires quasipolynomially many samples.

The main algorithm was obtained in a GPT-5.6 Sol conversation prompted by the author; the author is responsible for all statements, proofs, and correctness. Concurrent work of Boucher–Fouque–Shen gives an algorithm with similar guarantees for prime-power moduli.

Contents

1 Introduction **3**
- 1.1 Our main result and idea . . . 4
- 1.2 Related work . . . 4
- 1.3 Cryptographic non-implications . . . 8

2 Technical Overview **8**
- 2.1 Recalling the $q = 2^k$ algorithm . . . 9
- 2.2 The $q = 3^k$ case . . . 9
- 2.3 Generalizing to smooth q . . . 11

3 Preliminaries **11**
- 3.1 Subset sum over small field size . . . 12
- 3.2 Binomial basis for polynomials . . . 13
- 3.3 Group theory and characters . . . 13

4 Our algorithm **14**
- 4.1 Preparing Fourier samples . . . 15
- 4.2 A state combination procedure . . . 16
- 4.3 Modulus reduction . . . 17
- 4.4 Terminal recovery . . . 21
- 4.5 Putting everything together . . . 22

1 Introduction

The *dihedral coset problem* DCP_q^m asks to recover a secret $s \in \mathbb{Z}_q$ from m independent quantum samples of the form

$$\frac{1}{\sqrt{2}}\left(|0\rangle|x_i\rangle + |1\rangle|x_i + s\rangle\right), \qquad x_i \leftarrow \mathbb{Z}_q,\ i \in [m],$$

where the offsets x_i are uniformly random and unknown to the solver. States of this form naturally appear in the "standard approach" for the dihedral hidden subgroup problem and for the hidden shift problem over $\mathbb{Z}_q$ [Kup05, Reg04b]; solving DCP in polynomial time would imply polynomial-time quantum algorithms for both these problems. The best algorithm we know for all three of these problems is due to Kuperberg [Kup05], and has subexponential time and sample complexity $2^{O(\sqrt{\log q})}$.

Interestingly, DCP upper-bounds the hardness of the computational problems underlying several of our candidate post-quantum cryptographic assumptions. Key recovery for a cryptographic abelian group action is an instance of the hidden shift problem [CJS14], and Kuperberg-style DCP algorithms give the best known attacks for group action-based cryptography [BS20, Pei20]. Similarly, DCP is closely related to Learning With Errors (LWE), a central problem at the heart of lattice-based cryptography. Regev [Reg04a] showed that $\mathrm{poly}(n)$-uSVP (and therefore also LWE) is no harder than DCP, by giving a polynomial-time quantum reduction to DCP_q^m. Crucially, this reduction produces only polynomially many noiseless DCP samples, rendering Kuperberg's subexponential algorithm irrelevant. Information-theoretically, however, $m = O(\log q)$ samples already determine s [EHK04], and an algorithm with polynomial time- and sample-complexity for DCP would give a polynomial-time attack on lattice-based cryptography. The reduction, however, runs in one direction only: LWE might potentially be a lot easier than DCP.

To close this potential gap in hardness, Brakerski, Kirshanova, Stehlé and Wen [BKSW18] introduced the *extrapolated* dihedral coset problem $\mathsf{EDCP}_{n,q,f}^m$. It is parameterized additionally by a dimension $n \in \mathbb{N}$ and an amplitude function $f : \mathbb{Z} \to \mathbb{C}$ with $\sum_{k\in\mathbb{Z}} |f(k)|^2 = 1$, and asks to recover a secret *vector* $\mathbf{s} \in \mathbb{Z}_q^n$ from m independent quantum states of the form

$$\sum_{k\in\mathbb{Z}} f(k)|k\rangle|\mathbf{x}_i + k\cdot\mathbf{s}\rangle, \qquad \mathbf{x}_i \in \mathbb{Z}_q^n,\ i \in [m].$$

Taking $n = 1$ and $f(k) = \frac{1}{\sqrt{2}} \cdot \mathbb{1}[k \in \{0, 1\}]$ gives two-point DCP states. Writing $\mathsf{U\text{-}EDCP}_{n,q,r}^m$ for the *uniform* case, in which f is the normalized indicator of $\{0, 1, \ldots, r-1\}$, Brakerski et al. showed that $\mathsf{U\text{-}EDCP}_{n,q,r}^m$ and $\mathsf{LWE}_{n,q,\alpha}$ are equivalent up to polynomial losses in parameters under polynomial-time quantum reductions when $m = \mathrm{poly}(n \log q)$ and $r = \mathrm{poly}(n \log q)/\alpha$.

Throughout this paper, we specialize to the two-point amplitude $f(k) = \frac{1}{\sqrt{2}}\mathbb{1}[k \in \{0, 1\}]$, and call this problem *vector dihedral coset problem*, denoted as $\mathsf{DCP}_{n,q}^m$. This problem is known to be as hard as $\mathsf{U\text{-}EDCP}_{n,q,r}^m$ [BJK+25, Lemma 14]. Our goal is thus to recover $\mathbf{s}$ from independent samples of the form

$$\frac{1}{\sqrt{2}}\left(|0\rangle|\mathbf{x}_i\rangle + |1\rangle|\mathbf{x}_i + \mathbf{s}\rangle\right), \qquad \mathbf{x}_i \leftarrow \mathbb{Z}_q^n, i \in [m].$$

For general moduli, the best known algorithm is still Kuperberg's, which solves DCP_q^m in time and samples $2^{O(\sqrt{\log q})}$ [Kup05]; applied to the group $\mathbb{Z}_q^n$ it solves $\mathsf{DCP}_{n,q}^m$ in time $2^{O(\sqrt{n \log q})}$. For *structured* moduli, however, one can do considerably better. When $q = p$ is a fixed prime, Friedl, Ivanyos, Magniez, Santha and Sen [FIM+14] solve the problem in polynomial time, and when $q = 2^t$ is a power of two, Bai, Jangir, Kirshanova, Ngo and Youmans [BJK+25] give a quasipolynomial-time algorithm for $q = \mathrm{poly}(n)$. Both exploit special structure of the modulus, which raises the question:

For which moduli q does vector DCP admit an algorithm that beats Kuperberg's?

1.1 Our main result and idea

Call a modulus B-*smooth* if all of its prime factors are at most B. Our main result is an algorithm for $\mathsf{DCP}^m_{n,q}$ whose cost is governed by the largest prime factor of q.

Informal Theorem 1.1 (Corollary 4.18)**.** *Let $q > 1$ be B-smooth. There is a quantum algorithm that recovers $\mathbf{s} \in \mathbb{Z}_q^n$ from two-point vector DCP states with success probability at least $1 - \varepsilon$, using*

$$n^{O(B^{1+o(1)} \log q)} B^{O(B^{2+o(1)} \log q)} \operatorname{poly}\left(\log \frac{2}{\varepsilon}\right)$$

samples and time. This is quasipolynomial when $B = \operatorname{polylog}(n)$ and $q = \operatorname{poly}(n)$.

The work of Boucher, Fouque and Shen [BFS26] uses different techniques to obtain an algorithm for prime-power moduli $q = p^t$. We handle smooth moduli with possibly different prime factors. We compare our results in more detail in Section 1.2.

Our algorithm proceeds by reducing the modulus one prime factor $p|q$ at a time. The main idea is that this can be done by a sieving procedure that clears the non-linear terms in the phase modulo p, one degree at a time, via a subroutine that finds non-zero $\mathbf{u} \in \mathbb{F}_p^\ell$ such that $\sum_{i \in [\ell]} u_i^d \mathbf{a}_i = 0 \mod p$, given vectors $\mathbf{a}_1, \ldots, \mathbf{a}_\ell \in \mathbb{F}_p^n$. We use the diagonal-equation algorithm of Ivanyos and Santha [IS17] for this purpose; the subset sum algorithms of Imran and Ivanyos [II23] or Kothari–O'Donnell–Wu [KOW26] give alternative ways to find these relations with slightly worse dependence on B in the exponent of n. We describe our techniques in more detail in Section 2.

We describe our main algorithm (Algorithm 4.15) and state the main result (Theorem 4.16) in terms of DCP for the generalized dihedral group $\operatorname{dih}(G)$, where $\mathbb{Z}_q^n$ is replaced with a general finite abelian group G to which we have black-box access. This allows us to highlight the structure of $\mathbb{Z}_q^n$ that enables a faster algorithm (see Corollary 4.17).

1.2 Related work

The algorithm of Kuperberg [Kup05] gives a subexponential-time quantum algorithm for the *hidden shift problem* over arbitrary finite abelian groups: given oracle access to injective functions $f_0, f_1 : G \to X$ satisfying $f_0(x) = f_1(x + s)$ for some unknown $s \in G$, the goal is to recover s. Querying these functions in superposition and measuring the output register yields states of the form $\frac{1}{\sqrt{2}}(|0, x\rangle + |1, x + s\rangle)$, where $x \in G$ is uniformly random. These are coset states of the generalized dihedral group $\operatorname{dih}(G)$, defined in Definition 3.1.

Specializing to $G = \mathbb{Z}_q^n$ yields the vector dihedral coset problem, which Kuperberg's algorithm solves in time and sample complexity $2^{O(\sqrt{n \log q})}$, for arbitrary moduli q. In this setting, Kuperberg's algorithm works by first performing a quantum Fourier transform to convert DCP samples into Fourier samples of the form $\left(\mathbf{a}, \frac{1}{\sqrt{2}}(|0\rangle + \omega_q^{\langle \mathbf{a}, \mathbf{s} \rangle}|1\rangle)\right)$, and then performs a "sieving" step that iteratively combines these Fourier samples until $\mathbf{a} \in (q/2)\mathbb{Z}_q^n$. At this point, the least significant bits of $\mathbf{s}$ can be learned by a Hadamard transform and Gaussian elimination.

Memory-efficient algorithms. In addition to requiring subexponential samples and runtime, Kuperberg's original algorithm requires $2^{O(\sqrt{\log |G|})}$ quantum memory. Regev reduced the space complexity to $\mathrm{poly}(\log |G|)$ for cyclic 2-groups, at the expense of increasing the time and sample complexity to $2^{O(\sqrt{\log |G| \log\log |G|})}$ [Reg04b]. Kuperberg's later collimation sieve recovered the original $2^{O(\sqrt{\log |G|})}$ time bound while requiring only $O(\log |G|)$ qubits, at the cost of subexponential $2^{O(\sqrt{\log |G|})}$ classical memory accessible in quantum superposition [Kup13]. Peikert [Pei20] extended the collimation sieve to arbitrary cyclic groups and analyzed its concrete costs for CSIDH, an isogeny-based key-exchange protocol built from a commutative group action on supersingular elliptic curves.

Time- and sample-efficient algorithms for structured groups. While the preceding works reduce the space complexity of Kuperberg's algorithm, additional group structure can be exploited to get algorithms that run faster and use fewer samples. The simplest case is $G = \mathbb{Z}_2^n$, where hidden shift reduces to Simon's problem and can be solved in $\mathrm{poly}(n)$ time using $\mathrm{poly}(n)$ samples by Fourier sampling and then Gaussian elimination [Sim97]. Friedl, Ivanyos, Magniez, Santha, and Sen [FIM+14] generalized this result to $G = \mathbb{F}_p^n$ for any prime p, achieving time complexity $n^{O(p)} p^{O(p)}$. Their key observation is that quantum Fourier sampling can be used to obtain random linear *disequations* of the form $\langle a, s\rangle \neq 0 \bmod p$. By Fermat's little theorem, these can then be converted into linear equations in the degree-$(p-1)$ monomials of s, allowing the secret to be recovered using linear algebra [Iva08].

Friedl et al. [FIM+14] also extend their algorithm to more general groups using a recursion-friendly generalization of the hidden shift problem, in which the secret shift can be an entire coset, rather than just a single element.[1] Given hidden shift oracles $f_0, f_1 : G \to X$, they first prepare the states

$$|f_b\rangle := \frac{1}{\sqrt{|G|}} \sum_{x \in G} |x, f_b(x)\rangle,$$

so that the hidden shift relation becomes $s \cdot |f_0\rangle = |f_1\rangle$, where s acts by translating the first register by s; the recursive iterations of the algorithm prepare states with a coset of hidden shifts. Their recursive algorithm reduces this quantum-state version of hidden shift to instances over subgroups $H \leq G$ such that the quotient $G/H \cong \mathbb{F}_p^r$ is amenable to their $\mathbb{F}_p^n$ algorithm. In particular, for $G = \mathbb{Z}_q^n$ with all prime factors of q bounded by B, their algorithm runs in time $(n \log q)^{O_B(\log q)}$, which is quasipolynomial for constant B and $q = \mathrm{poly}(n)$.[2] However, their recursion uses copies of the full function states $|f_0\rangle, |f_1\rangle$ and coherent operations on them, rather than just independent dihedral coset states. It therefore does not directly yield an algorithm for the vector DCP problem considered here.[3]

Our algorithm can be viewed as a counterpart to the hidden shift algorithm of Friedl et al. [FIM+14] that works directly with independent DCP samples, rather than coherent access to hidden shift function oracles. The two approaches also use complementary classical subroutines: Friedl et al. solve linear disequations

[1] Friedl et al. call this generalization the Translating Coset problem.

[2] They study hidden shift over nonabelian groups as well, and refer to this variant as the *hidden translation* problem. They obtain polynomial-time algorithms for smoothly solvable groups, quasipolynomial-time algorithms for solvable groups of constant exponent, and subexponential-time algorithms for arbitrary solvable groups. While it is not important for this work, a group G is *solvable* if it admits a subnormal series $G = G_0 \geq G_1 \geq \ldots \geq \{0\}$ with abelian quotients G_i/G_{i+1}, and its *exponent* is the least common multiple of the orders of its elements.

[3] We remark that this distinction does not rule out cryptographic applications: group-action-based cryptosystems provide coherent access to the hidden shift functions, and adaptations of Regev's reduction may yield noisy versions of the function states required by Friedl et al. Nevertheless, this recursive algorithm in Friedl et al.'s work appears to have received relatively little attention in the cryptographic literature, possibly because it is formulated in terms of Translating Coset rather than DCP.

over $\mathbb{F}_p^n$, whereas we solve diagonal equations over $\mathbb{F}_p^n$, with subset sum as a special case. This suggests a duality between their approach and ours. The duality of the two problems can also be seen in the work of Chen, Liu, and Zhandry [CLZ22], which reduces subset sum over $\mathbb{F}_p^n$ to solving disequations via Fourier transform and unambiguous state discrimination.

Algorithms for DCP over power-of-two moduli. For $q = 2^k$, faster algorithms are known that work directly with independent DCP states, achieving time and sample complexity $n^{O(\log q)}$. Bonnetain and Naya-Plasencia [BNP18] introduced the *Simon-meets-Kuperberg* algorithm, which interpolates between Simon's polynomial-time algorithm when $k = 1$ and Kuperberg's subexponential-time algorithm when $n = 1$. Csáji [Csá21] subsequently gave a rigorous, space-efficient algorithm achieving $n^{O(\log q)}$ time and sample complexity. More recently, Bai, Jangir, Kirshanova, Ngo, and Youmans [BJK+25] gave a provable variant of Simon-meets-Kuperberg that also handles extrapolated dihedral coset states (EDCP), achieving the same asymptotic complexity for power-of-two moduli. These algorithms exploit linear dependencies over $\mathbb{F}_2$ to progressively reduce the modulus by factors of two.

Beyond powers of two. Castryck, Dooms, Emerencia, and Lemmens [CDEL21] combined the power-of-two sieving techniques of [BNP18] with the $\mathbb{F}_p^n$ algorithm of Friedl et al. [FIM+14], to give a $n^{O_p(\log q)}$ algorithm for vector DCP over moduli of the form $q = p \cdot 2^k$. They also extend Peikert's collimation sieve to arbitrary finite abelian groups. Combining these techniques, they obtain a hidden shift algorithm for any finite abelian group G running in time $\mathrm{poly}(\log |G|) \cdot 2^{O(\sqrt{\log |2^k p G|})}$, using only polynomially many qubits and quantum-accessible classical memory.

Concurrent work of Boucher, Fouque, and Shen [BFS26]. The recent concurrent work by Boucher, Fouque, and Shen [BFS26] extends the Simon-meets-Kuperberg framework to vector DCP over prime-power moduli $q = p^k$, achieving $n^{O_p(\log q)}$ time and sample complexity. They do this by (1) defining the *cyclotomic coset problem* (CCP), a p-ary generalization of the binary *dihedral* coset problem (DCP), (2) generalizing the Simon-meets-Kuperberg framework to CCP, and (3) giving a reduction from DCP to CCP.

The dihedral group $\mathrm{dih}(\mathbb{Z}_q^n)$ can be viewed as a semidirect product $\mathbb{Z}_q^n \rtimes \mathbb{Z}_2$, with group law $(\mathbf{x}, a) \times (\mathbf{y}, b) = (\mathbf{x} + (-1)^a \mathbf{y}, a + b)$, and DCP asks to recover the subgroup $\{(0,0), (\mathbf{s}, 1)\}$ from the coset states $\frac{1}{\sqrt{2}}(|0\rangle|\mathbf{x}\rangle + |1\rangle|\mathbf{x} + \mathbf{s}\rangle)$. Boucher, Fouque, and Shen [BFS26] define CCP analogously for the cyclotomic group $\mathbb{Z}_q[\zeta_p]^n \rtimes \mathbb{Z}_p$, with group law

$$(\mathbf{x}, a) \times (\mathbf{y}, b) = (\mathbf{x} + \zeta_p^a \mathbf{y}, a + b),$$

where ζ_p is a primitive pth root of unity. They define CCP to be the task of recovering the subgroup $\{(\lambda_a \cdot \mathbf{s}, a) : a \in \mathbb{Z}_p\}$, where $\lambda_a := \sum_{j=0}^{a-1} \zeta_p^j$, from coset states of the form

$$\frac{1}{\sqrt{p}} \sum_{a \in \mathbb{Z}_p} |a\rangle|\mathbf{x} + \lambda_a \cdot \mathbf{s}\rangle.$$

They then extend the Simon-meets-Kuperberg framework to work for CCP states, via a subset sum algorithm over $\mathbb{F}_p^n$ of [II23].

Our algorithm also combines Fourier samples using relations among their labels. We work directly with the Fourier samples obtained from the DCP coset states, clearing nonlinear phase terms modulo p one degree at a time to get "semi-clean" states whose phase terms are linear modulo p. Such "semi-clean"

Fourier samples can be used to reduce the modulus by a factor of p; this process can then be repeated. In the language of our algorithm, the cyclotomic states of [BFS26] are already semi-clean modulo $\zeta_p - 1$: their degree-d binomial coefficient is $(\zeta_p - 1)^{d-1} y$, so all nonlinear coefficients vanish modulo $\zeta_p - 1$, whereas our algorithm achieves this linearity by explicitly cancelling the nonlinear coefficients of arbitrary phase-label maps.

Although their current analysis is restricted to prime power moduli, we suspect that it might be possible to extend the sieve of [BFS26] to smooth moduli, perhaps by giving a reduction that prepares coset states for one cyclotomic group $\mathbb{Z}_q[\zeta_{p_1}]^n \rtimes \mathbb{F}_{p_1}$ from coset states for another cyclotomic group $\mathbb{Z}_q[\zeta_{p_2}]^n \rtimes \mathbb{F}_{p_2}$, where $p_1, p_2 | q$, but $p_1 \neq p_2$; however this remains to be checked. Our construction directly works with p-ary Fourier samples, and handles the more general smooth modulus setting.

While [BFS26] uses the subset sum algorithm of [II23], we observe that by instead using the algorithm of Ivanyos and Santha [IS17] to solve the more general degree-d problem of finding nonzero $\mathbf{u}$ such that $\sum_i u_i^d \mathbf{a}_i = \mathbf{0}$, we get a slightly better dependence on B in the exponent of n in the sample and time complexity. For $q = p^t$, our time and sample bound from Corollary 4.18 is

$$n^{O(p^{1+o(1)} \log q)} p^{O(p^{2+o(1)} \log q)}.$$

The theorem of [BFS26] states a $2^{O_p(\log n \log q)}$ bound for fixed p, but to the best of our understanding, making the dependence on p explicit gives a $n^{O(p^2 \log q)} p^{O(p^2 \log p \log q)}$ algorithm.

Diagonal polynomial equations and subset sum over $\mathbb{F}_p^n$. Our algorithm needs a subroutine that, given vectors $\mathbf{a}_1, \ldots, \mathbf{a}_\ell \in \mathbb{F}_p^n$ and a degree $1 \leq d \leq p-1$, finds a nonzero $\mathbf{u} \in \mathbb{F}_p^\ell$ satisfying $\sum_{i=1}^\ell u_i^d \mathbf{a}_i = \mathbf{0}$. Such a relation lets us combine Fourier samples so that their degree-d coefficient vanishes, and we do this for degrees $d = p-1$ through 2. Ivanyos and Santha [IS17, Section 3] give a deterministic algorithm for this task. Writing $g = \gcd(d, p-1)$, a sufficient number of arbitrary input vectors is $\ell = n^{O(g \log(g+1))} \cdot g^{O(g^2 \log(g+1))}$, and the running time is $\mathrm{poly}(\ell, p)$; see Theorem 3.4. This is the subroutine we use.

We observe that a subset sum algorithm also suffices: it finds a nonzero $\mathbf{u} \in \{0,1\}^\ell$ with $\sum_i u_i \mathbf{a}_i = \mathbf{0}$. Since $u_i^d = u_i$ for these coefficients, the same relation solves the degree-d problem for every d.

Imran and Ivanyos [II23, Theorem 2] give a deterministic subset sum algorithm using $\ell = n^{O(p \log p)} \cdot p^{O(p \log^2 p)}$ arbitrary input vectors and $\mathrm{poly}(\ell)$ time. They use it as a subroutine to get a polynomial-time HSP algorithm for nilpotent groups of constant class whose orders have bounded prime factors [II23, Theorem 1]. The subset sum routine is also used to reduce the problem to a quotient group, however their work does not cover our dihedral groups: $\mathrm{dih}(\mathbb{Z}_q^n)$ is not nilpotent when q has an odd prime factor, and for $q = 2^k$ its nilpotency class is k, which can grow. Their exact theorem also assumes access to a unitary preparing a purification of the subgroup state and its inverse; independent DCP samples do not supply this access, although their central-subgroup conversion itself can use coset-state copies.

Kothari, O'Donnell, and Wu [KOW26, Sections 3.2 and 5.2] improve two steps of Imran–Ivanyos's method. For $p \geq 5$, their subset sum algorithm for independent and uniformly random inputs requires only $\ell = n^{2\lfloor (p+3)/4 \rfloor} p^{O(p^2)}$ inputs. For $p = 3$, Kothari–O'Donnell–Wu gives a deterministic algorithm using $O(n^2)$ arbitrary inputs; see Lemma 3.3.

More recently, Kothari, Metger, O'Donnell, Shutty, and Wu [KMO+26] gave a polynomial-time quantum algorithm using εn^2 independent uniform inputs over $\mathbb{F}_3^n$ for any fixed $\varepsilon > 0$, and improved the polynomial-time classical bound over $\mathbb{F}_p^n$, for $p \geq 5$, to $p^{O(p^2)} n^{1+\lfloor (p+3)/4 \rfloor}$ independent uniform inputs.

1.3 Cryptographic non-implications

The equivalence of LWE and suitable forms of EDCP does not make the present algorithm an attack on standard LWE. The reductions of Brakerski et al. [BKSW18] produce only a polynomial number of clean EDCP states, while our algorithm uses a quasipolynomial number. An LWE consequence would therefore require either a polynomial-sample version of our algorithm or a reduction that supplies the larger collection of suitably clean states it needs. The same sample barrier was noted for the power-of-two algorithm of Bai et al. [BJK+25].

AI disclosure. The main result was found by ChatGPT-5.6 Sol in a conversation with the author in July 2026, except for the optimization to use the [IS17] algorithm, which was found in conversation with ChatGPT-6 Astra on 26 September 2026. The author verified and simplified the algorithm, over several conversations with ChatGPT-5.6 Sol, Claude Fable 5.1, ChatGPT-6 Sol and Astra; the author takes full responsibility for correctness, exposition, and attribution.

The above models were also used to help write and edit this manuscript. The introduction and technical overview were primarily written by the author, with AI assistance mainly for copy-editing. In the remaining sections, the author used AI tools over several rounds to expand initial drafts, obtain feedback, and revise the exposition until the author was satisfied with its presentation.

Acknowledgements. The author thanks Vinod Vaikuntanathan, Noah Shutty, and Seyoon Ragavan for useful discussions. We also thank an anonymous reviewer for pointing out that our quasipolynomial sample and runtime guarantees hold even for smoothness bound $B = \mathrm{polylog}(n)$, not just when $B = O(1)$. The author was supported in part by a Simons investigator award and NSF grant CNS-2534400, and thanks the UK AISI alignment project for supporting access to ChatGPT Pro.

2 Technical Overview

We recall that our goal is to solve two-point vector EDCP: we are given samples

$$\frac{1}{\sqrt{2}}\left(|0\rangle|\mathbf{x}_i\rangle + |1\rangle|\mathbf{x}_i + \mathbf{s}\rangle\right),$$

where $\mathbf{s} \in \mathbb{Z}_q^n$ is a fixed unknown secret and each $\mathbf{x}_i \sim \mathbb{Z}_q^n$, $i \in [m]$ is uniform and unknown, and our goal is to find $\mathbf{s}$. A standard first step is Fourier sampling: measuring the second register of a sample in the q-ary Fourier basis returns a uniformly random label $\mathbf{a} \in \mathbb{Z}_q^n$ and leaves, up to global phase, the single-qubit phase state

$$\frac{1}{\sqrt{2}}\left(|0\rangle + \omega_q^{\langle \mathbf{a},\mathbf{s}\rangle}|1\rangle\right), \tag{1}$$

where $\omega_q := e^{2\pi i/q}$. We call this a *binary Fourier sample* with label $\mathbf{a}$; the label $\mathbf{a}$ is known to the algorithm but the secret $\mathbf{s}$ is not. A single phase qubit is of little use on its own, because the phase $\omega_q^{\langle \mathbf{a},\mathbf{s}\rangle}$ cannot be read off by a measurement in general. The $q = 2$ case is exactly Simon's algorithm [Sim97]: performing a Hadamard measurement returns $\langle \mathbf{a}, \mathbf{s}\rangle \bmod 2$ exactly, and $O(n)$ such equations with independent uniform labels determine $\mathbf{s}$.

Now let us think about the case of larger q; assume for now that it is even. If we had gotten lucky, and it so happened that the measurement resulted in $\mathbf{a} \in (q/2) \cdot \mathbb{Z}_q^n \cong \mathbb{Z}_2^n$, we would know that $\langle \mathbf{a}, \mathbf{s}\rangle$ lies in

$\{0, q/2\}$, and the same Hadamard-basis measurement would allow us to learn $\left(\frac{2}{q}\langle \mathbf{a}, \mathbf{s}\rangle\right) \bmod 2$. The high-level idea of Kuperberg's sieve [Kup05] is to combine binary Fourier samples so that their labels become progressively "more divisible" by $q/2$, until the phases can be read off by a Hadamard measurement.

The Simon-meets-Kuperberg algorithms [BNP18, Csá21, BJK+25] implement this strategy particularly efficiently for power-of-two moduli, using linear algebra over $\mathbb{F}_2$ to reduce the modulus by a factor of 2 at each step. As we will see, this more efficient state-combination procedure does not directly extend to odd primes; our contribution is to overcome this obstacle for smooth moduli.

2.1 Recalling the $q = 2^k$ algorithm

It will be instructive for us to first recall the Simon-meets-Kuperberg algorithms of [BNP18, BJK+25] for the special case when $q = 2^k$ for some $k \in \mathbb{N}$. Take $n+1$ phase qubits with labels $\mathbf{a}_1, \ldots, \mathbf{a}_{n+1}$ and collect the labels as the rows of a known matrix $\mathbf{A} \in \mathbb{Z}_q^{(n+1)\times n}$. The tensor product of these $n+1$ states is

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{\mathbf{x}\in\{0,1\}^{n+1}} \omega_q^{\langle \mathbf{A}^\top \mathbf{x}, \mathbf{s}\rangle} |\mathbf{x}\rangle, \tag{2}$$

where we interpret $\mathbf{x} \in \mathbb{Z}_q^{n+1}$ in the phase term. We would like to retain only those $\mathbf{x}$ which are in the mod 2 kernel of $\mathbf{A}^\top$, i.e., with $\mathbf{A}^\top \mathbf{x} \equiv 0 \bmod 2$, since those are exactly the branches whose phase labels lie in $2\mathbb{Z}_q^n \cong \mathbb{Z}_{q/2}^n$. A naive hope might be to coherently measure a bit indicating whether $\mathbf{x} \in \ker(\mathbf{A}^\top \bmod 2)$ and postselect on success, but the set of such $\mathbf{x}$ is far too sparse to do this—there are exactly two such vectors with good probability, namely $\mathbf{0}$ and the nonzero $\mathbf{u} \in \mathbb{F}_2^{n+1}$ kernel vector of $\mathbf{A}^\top \bmod 2$.

The key observation is that any single *coset* of the kernel is just as good. Unlike the kernel itself, its cosets can be filtered for as follows: we first parameterize $\mathbf{x} = (\mathbf{z} + t\mathbf{u})$ with $\mathbf{z}$ ranging over fixed representatives of the cosets in $\mathbb{F}_2^{n+1}/\langle \mathbf{u}\rangle$ and $t \in \mathbb{F}_2$ so that Equation (2) reads

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{\substack{\mathbf{z}\in\mathbb{F}_2^{n+1}/\langle \mathbf{u}\rangle, \\ t\in\mathbb{F}_2}} \omega_q^{\langle \mathbf{A}^\top((\mathbf{z}+t\mathbf{u}) \bmod 2), \mathbf{s}\rangle} |t\rangle|\mathbf{z}\rangle.$$

Measuring the second register—that is, measuring which coset of $\langle \mathbf{u}\rangle$ the branch lies in—leaves, up to global phase, a binary Fourier sample $\frac{1}{\sqrt{2}}(|0\rangle + \omega_{q/2}^{\langle \mathbf{a}', \mathbf{s}\rangle}|1\rangle)$, with uniformly random and known label

$$\mathbf{a}' := \frac{\mathbf{A}^\top\big(((\mathbf{z}+\mathbf{u}) \bmod 2) - \mathbf{z}\big)}{2} \in \mathbb{Z}_{q/2}^n.$$

Iterating this procedure brings the modulus down from $q = 2^k$ to 2, at a cost of $n+1$ phase qubits per level and hence $n^{O(k)}$ per modulus 2 phase qubit. A Hadamard measurement at the very end then gives one equation $\langle \tilde{\mathbf{a}}, \mathbf{s}\rangle \equiv c \bmod 2$ with a known uniform $\tilde{\mathbf{a}}$, and $O(n)$ of them determine $\mathbf{s} \bmod 2$ by Gaussian elimination. Once this last bit of the secret is known it can be stripped off and the whole procedure can be repeated on the next bit. Therefore this algorithm has sample and time complexity $n^{O(\log q)}$, which is quasipolynomial when $q = \mathrm{poly}(n)$.

2.2 The $q = 3^k$ case

We could try to peel off one factor of 3 at each level and use the Fourier transform over $\mathbb{F}_3$ at the end to learn an equation in $\mathbf{s}$. But a binary Fourier sample still has only two branches, whereas an exact Fourier measurement modulo 3 requires a three-branch Fourier state. Moreover, if we measure $\mathbf{A}^\top \mathbf{x} \bmod 3$ on m binary samples, the surviving set need not be a coset of the kernel, and need not have three elements.

From qubits to qutrits. The mismatch between the support size of the states, which is 2, and the prime we wish to divide out, which is 3, motivates converting these binary qubits into qutrits supported on 3 basis states. Given two Fourier samples with labels $\mathbf{a}, \mathbf{a}'$,

$$\frac{1}{2}\left(|00\rangle + \omega_q^{\langle \mathbf{a}', \mathbf{s}\rangle}|01\rangle + \omega_q^{\langle \mathbf{a}, \mathbf{s}\rangle}|10\rangle + \omega_q^{\langle \mathbf{a}+\mathbf{a}', \mathbf{s}\rangle}|11\rangle\right),$$

we can project the joint state onto $\mathrm{span}(|00\rangle, |10\rangle, |11\rangle)$, which succeeds with probability $3/4$. Relabeling $|00\rangle \mapsto |0\rangle$, $|10\rangle \mapsto |1\rangle$, $|11\rangle \mapsto |2\rangle$, and writing $\mathbf{b} := \mathbf{a}' - \mathbf{a}$, gives the resulting qutrit

$$\frac{1}{\sqrt{3}}\left(|0\rangle + \omega_q^{\langle \mathbf{a}, \mathbf{s}\rangle}|1\rangle + \omega_q^{\langle 2\mathbf{a}+\mathbf{b}, \mathbf{s}\rangle}|2\rangle\right).$$

Since $(\mathbf{a}, \mathbf{a}') \mapsto (\mathbf{a}, \mathbf{b})$ is invertible, the labels $\mathbf{a}, \mathbf{b}$ are again independent and uniform. It is convenient to record such a qutrit by a function corresponding to its phase terms $f : \mathbb{Z}_3 \to \mathbb{Z}_q^n$, defined by $f(0) = 0$, $f(1) = \mathbf{a}$, and $f(2) = 2\mathbf{a} + \mathbf{b}$. That is,

$$f(x) \equiv \mathbf{a}x + \mathbf{b}\binom{x}{2} \pmod 3.$$

The above form is the expansion of f in the binomial basis $\binom{x}{1}, \binom{x}{2}$, which will be a convenient coordinate system for maps out of $\mathbb{Z}_p$ (see Lemma 3.5). Call a qutrit *semi-clean* if its quadratic coefficient vanishes, that is, $\mathbf{b} \equiv 0 \bmod 3$. For a semi-clean qutrit the label map is linear modulo 3, and the binary argument goes through verbatim: combining $n+1$ semi-clean qutrits and measuring the coset of a kernel direction produces a qutrit whose labels are all divisible by 3, so the modulus drops from q to $q/3$. Therefore the problem has been reduced to manufacturing semi-clean qutrits, or equivalently, clearing the quadratic coefficient $\mathbf{b}$.

Clearing the quadratic coefficient $\mathbf{b}$ via $\mathbb{F}_3^n$-subset sum [II23]. Take m qutrits with phase term functions $f_i(x) = \mathbf{a}_i x + \mathbf{b}_i\binom{x}{2}$. Their tensor product is

$$\frac{1}{\sqrt{3^m}} \sum_{\mathbf{x}\in\mathbb{F}_3^m} \omega_q^{\langle g(\mathbf{x}), \mathbf{s}\rangle}|\mathbf{x}\rangle, \qquad g(\mathbf{x}) := \sum_{i=1}^m f_i(x_i) \equiv \sum_{i=1}^m \left(\mathbf{a}_i x_i + \mathbf{b}_i\binom{x_i}{2}\right) \pmod 3.$$

We could try to apply a similar state-combination procedure as the power-of-two algorithm, and for some fixed nonzero vector $\mathbf{u} \in \mathbb{F}_3^m$, we measure the coset of $\mathbf{x}$ modulo $\langle \mathbf{u}\rangle$. It is not clear yet what $\mathbf{u}$ should be; we will see a natural choice shortly. This measurement leaves a uniform superposition over an affine line $\mathbf{x} = \mathbf{z} + t\mathbf{u}$, $t \in \mathbb{F}_3$ for some measurement outcome $\mathbf{z}$ known to the algorithm. Restricted to that line, $g \bmod 3$ becomes a univariate map of degree at most 2,

$$g(\mathbf{z} + t\mathbf{u}) \equiv \mathbf{c}_0 + \mathbf{c}_1 t + \mathbf{c}_2\binom{t}{2} \pmod 3.$$

A simple calculation (see Lemma 3.6) shows that its quadratic coefficient is $\mathbf{c}_2 = \sum_i u_i^2 \mathbf{b}_i \bmod 3$. Because $u_i^2 = \mathbb{1}[u_i \neq 0]$ in $\mathbb{F}_3$, this coefficient depends only on the support of $\mathbf{u}$, and it vanishes exactly when

$$\sum_{i\,:\,u_i\neq 0} \mathbf{b}_i \equiv \mathbf{0} \pmod 3.$$

Finding such a vector $\mathbf{u}$ is an instance of subset-sum over $\mathbb{F}_3^n$, given input $\mathbf{b}_1, \ldots, \mathbf{b}_m$. If $m = \Theta(n^2)$, there is a polynomial-time algorithm that finds such a solution, due to [II23, KOW26].

One semi-clean qutrit costs $O(n^2)$ qutrits, and one modulus reduction costs $n+1$ semi-clean qutrits, so a single level costs $O(n^3)$ input states. Running $\log_3 q$ levels and collecting $O(n)$ equations per digit gives sample and time complexity $n^{O(\log q)}$, which is quasipolynomial for $q = \mathrm{poly}(n)$.

2.3 Generalizing to smooth q

The techniques for the $q = 3^k$ case can be readily generalized to smooth moduli.

First, we can use a method similar to how we prepared qutrits (or 3-ary Fourier samples) from binary Fourier samples, to also prepare p-ary Fourier samples (and vice versa).

As in the power-of-three algorithm, we reduce these labels modulo p and expand in the binomial basis, giving

$$f(t) \equiv \sum_{d=1}^{p-1} \binom{t}{d} \mathbf{a}^{(d)} \pmod{p}.$$

We clear the coefficients from degree $p-1$ down to degree 2. At degree d, collapsing a collection of states to a coset of $\mathbf{u}$ changes the degree-d coefficient to

$$\sum_i u_i^d \mathbf{a}_i^{(d)}. \tag{3}$$

Since we want to reduce the degree modulo p, we choose $\mathbf{u}$ to make this sum zero. Higher degrees do not reappear, so this results in a p-ary Fourier sample where the phase terms are described by a degree-$(d-1)$ function modulo p.

When $p = 3$ and $d = 2$, this corresponds exactly to subset sum over $\mathbb{F}_3^n$. For every p and d, a subset-sum solution $\mathbf{u} \in \{0,1\}^m$ also suffices, since $u_i^d = u_i$. Such a solution can be supplied by Imran–Ivanyos [II23] (or Kothari–O'Donnell–Wu [KOW26], but we additionally must prove that the input vectors remain uniformly random). Using the Imran–Ivanyos bound requires $m = n^{O(p \log p)} p^{O(p \log^2 p)}$ inputs per degree. The costs multiply over the degrees $p-1, \ldots, 2$, so using only this subset sum routine would cost $n^{O(p^2 \log p)} p^{O(p^2 \log^2 p)}$ samples per modulus reduction.

However, we do not need a subset-sum solution $\mathbf{u} \in \{0,1\}^m$: any nonzero $\mathbf{u} \in \mathbb{F}_p^m$ making Equation (3) zero will do. Writing $g = \gcd(d, p-1)$, the Ivanyos–Santha algorithm finds such a relation given

$$\ell(p, n, d) \geq n^{O(g \log p)} p^{O(g^2 \log p)}$$

inputs. Since $g \ll p$ often, this gives us a better dependence on B in the exponent of n: for each divisor $g \mid (p-1)$, at most $(p-1)/g$ degrees have $\gcd(d, p-1) = g$, since each is a multiple of g. Thus the contribution to the exponent of n from each divisor is at most $\frac{p}{g} O(g \log p) = O(p \log p)$. Since $p-1$ has $p^{o(1)}$ divisors by Lemma 3.11, the product over all degrees is at most

$$n^{O(p^{1+o(1)})} p^{O(p^{2+o(1)})}.$$

3 Preliminaries

Notation and terminology. We write $\mathbb{Z}_q$ to mean $\mathbb{Z}/q\mathbb{Z}$.

For an abelian group G, a subgroup $H \leq G$, and $g \in G$, we write $g + H \in G/H$ for the quotient class and $[g]_H \in G$ for a fixed canonical representative of that class. For a function f valued in G, we use these notations pointwise: $(f + H)(x) = f(x) + H$ and $[f]_H(x) = [f(x)]_H$.

Throughout this paper, we identify quantum states up to global phase.

Definition 3.1. *Let G be a finite abelian group, written additively. The **generalized dihedral group** of G is $\mathrm{dih}(G) := G \rtimes_{-1} \mathbb{Z}_2$, where the nontrivial element of $\mathbb{Z}_2$ acts on G by inversion. That is, writing elements in $\mathrm{dih}(G)$ as pairs $(b, x) \in \mathbb{Z}_2 \times G$, the group operation is*

$$(b_1, x_1) \cdot (b_2, x_2) = (b_1 + b_2,\ x_1 + (-1)^{b_1} x_2).$$

The dihedral group of $2q$ elements is obtained by taking $G = \mathbb{Z}_q$.

Definition 3.2. *Let G be a finite abelian group, and let $s \in G$ be a fixed secret. The generalized dihedral coset problem asks to recover s from samples of the form*

$$\frac{1}{\sqrt{2}} \left(|0\rangle |x_i\rangle + |1\rangle |x_i + s\rangle \right),$$

where $x_i \sim G$ are uniformly random and unknown.

In this work, we are especially interested in the vector dihedral coset problem, which specializes $G = \mathbb{Z}_q^n$.

3.1 Subset sum over small field size

Lemma 3.3 (Imran–Ivanyos [II23] and Kothari–O'Donnell–Wu [KOW26])**.** *Let p be prime and $n \geq 1$. The following algorithms find a nonempty zero-sum subset of vectors in $\mathbb{F}_p^n$.*

1. *Theorem 2 of Imran and Ivanyos gives a deterministic algorithm for any $\ell \geq p^{O(p \log^2 p)} n^{O(p \log p)}$ input vectors, running in time $\mathrm{poly}(\ell)$.*
2. *For $p = 3$ and $n \geq 2$, Theorem 1.2 of Kothari, O'Donnell, and Wu gives a deterministic algorithm, for any $\ell \geq n^2/3 + Cn \log n$ input vectors, running in time $\mathrm{poly}(\ell)$, for an absolute constant C.*
3. *For $p \geq 5$, Theorem 4.12 of Kothari, O'Donnell, and Wu gives a deterministic algorithm that succeeds with probability at least $2/3$ on $\ell \geq p^{O(p^2)} n^{2\lfloor (p+3)/4 \rfloor}$ independent uniform input vectors and runs in time $\mathrm{poly}(\ell)$.*

These subset sum routines give alternative ways to find the relations used below. Our algorithm instead uses the following Ivanyos–Santha theorem at every degree.

Theorem 3.4 (Ivanyos–Santha [IS17, Section 3])**.** *Let p be prime, $n \geq 1$, and $1 \leq d < p$, and put $g = \gcd(d, p-1)$. Given any ℓ vectors $\mathbf{a}_1, \ldots, \mathbf{a}_\ell \in \mathbb{F}_p^n$ with*

$$\ell \geq (n+1)^{g \lceil \log_2(g+1) \rceil} g^{\frac{g(g-1)}{2} \lceil \log_2(g+1) \rceil},$$

the deterministic algorithm of Ivanyos and Santha runs in time $\mathrm{poly}(\ell, p)$ and finds a nonzero $\mathbf{u} \in \mathbb{F}_p^\ell$ such that $\sum_{i=1}^{\ell} u_i^d \mathbf{a}_i = \mathbf{0}$.

3.2 Binomial basis for polynomials

Lemma 3.5. *Let p be prime. Every map $g : \mathbb{Z}_p \to \mathbb{Z}_p^n$ satisfying $g(0) = 0$ has a unique expression*

$$g(x) = \sum_{d=1}^{p-1} \binom{x}{d} \mathbf{a}^{(d)}, \qquad \mathbf{a}^{(d)} = \sum_{x=0}^{d} (-1)^{d-x} \binom{d}{x} g(x).$$

In particular, if $g(1), \ldots, g(p-1)$ are independent and uniform in $\mathbb{Z}_p^n$, then $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(p-1)}$ are also independent and uniform in $\mathbb{Z}_p^n$.

Proof. The matrix with entries $\binom{x}{d}$ for $x, d = 1$ to $p-1$ is lower triangular with every diagonal entry equal to one, and is therefore invertible over $\mathbb{Z}_p$. The displayed inverse is the usual finite difference formula. The final assertion follows because an invertible linear map preserves the uniform distribution. □

Lemma 3.6. *Suppose*

$$g(x) = \sum_{j=1}^{d} \binom{x}{j} \mathbf{a}^{(j)} \pmod{p}$$

for some $d < p$. Then $g(z + tu) - g(z)$ has degree at most d in t, and its degree-d coefficient in the binomial basis is $u^d \mathbf{a}^{(d)}$. Its linear coefficient is $u\mathbf{a}^{(1)}$ plus a value determined by $\mathbf{a}^{(2)}, \ldots, \mathbf{a}^{(d)}, z$, and u.

Proof. The polynomial $\binom{x}{d}$ has degree d and leading coefficient $1/d!$. Substituting $x = z + tu$ multiplies its leading coefficient by u^d. Since the degree-d binomial polynomial in t also has leading coefficient $1/d!$, its coefficient after substitution is u^d. Terms of smaller degree cannot contribute to it. Finally, the linear input term contributes

$$(z + tu)\mathbf{a}^{(1)} - z\mathbf{a}^{(1)} = tu\mathbf{a}^{(1)},$$

and every other contribution to the linear coefficient comes from the coefficients of degrees 2 to d. □

3.3 Group theory and characters

Lemma 3.7. *Every finite abelian group G is isomorphic to a direct sum of cyclic groups,*

$$G \cong (\mathbb{Z}/p_1^{\alpha_1}\mathbb{Z})^{n_1} \times \cdots \times (\mathbb{Z}/p_k^{\alpha_k}\mathbb{Z})^{n_k}$$

Definition 3.8. *A character of a finite abelian group G is a homomorphism $\chi : G \to \mathbb{C}$. The dual group $\widehat{G}$ is the set of these characters, where the group law is pointwise multiplication, i.e.,*

$$(\chi_1\chi_2)(g) := \chi_1(g) \cdot \chi_2(g).$$

Lemma 3.9. *Let G be a finite abelian group. Then, $\widehat{G} \cong G$. Further, $\widehat{\widehat{G}}$ is canonically isomorphic to G, that is, there is an isomorphism $G \to \widehat{\widehat{G}}$ that can be defined without fixing a generator.*

We write this canonical isomorphism as $g \to \chi_g$, for $g \in G$.

Definition 3.10. *For $H \leq G$, its annihilator in $\widehat{G}$ is*

$$H^{\perp} = \{\chi \in \widehat{G} : \chi(h) = 1 \text{ for every } h \in H\}.$$

Lemma 3.11. *Let $N \geq 1$ be an integer. Then the number of positive divisors of N satisfies*

$$\#\{g \geq 1 : g \mid N\} = N^{o(1)}.$$

Proof. We will show that for every fixed constant $c > 0$, we have that

$$\#\{g \geq 1 : g \mid N\} = O(N^c) \qquad (N \to \infty).$$

Let $c > 0$ be an arbitrary constant. Write $N = \prod_{i=1}^{k} p_i^{\alpha_i}$, where the p_i are distinct primes and $\alpha_i \geq 1$. Then, we have that

$$\#\{g \geq 1 : g \mid N\} = \prod_{i=1}^{k} (\alpha_i + 1).$$

If $p_i^c \geq 2$, then $\alpha_i + 1 \leq 2^{\alpha_i} \leq p_i^{c\alpha_i}$. Only finitely many primes satisfy $p_i^c < 2$, and for each of them

$$\sup_{\alpha_i \geq 1} (\alpha_i + 1) p_i^{-c\alpha_i} < \infty$$

because $p_i^c > 1$. Multiplying these bounds gives $\prod_{i=1}^{k} (\alpha_i + 1) = O(N^c)$. □

4 Our algorithm

In this section, we present our algorithm and its proof of correctness for the dihedral coset problem over $\mathrm{Dih}(G)$, for a general abelian group G equipped with the necessary computational subroutines. We later specialize to the case $G = (\mathbb{Z}/q\mathbb{Z})^n$.

Black-box group model. We assume that we have black-box access to the group, i.e., every group element is identified by a unique string (of length $O(\log |G|)$, say), and we are given access to an oracle that performs the group operation. Further, we are given a set of generators for G. Cheung and Mosca [CM01] give an algorithm that, given black-box access to G as described above, computes a cyclic decomposition of G as in Lemma 3.7. Therefore, we assume without loss of generality that we have efficient algorithms that compute the forward and inverse maps for the cyclic decomposition isomorphism

$$G \cong (\mathbb{Z}/p_1^{\alpha_1}\mathbb{Z})^{n_1} \times \cdots \times (\mathbb{Z}/p_k^{\alpha_k}\mathbb{Z})^{n_k}.$$

This allows us to work with quotient groups and characters of the group easily, so for our purposes, we are effectively working with the explicit cyclic decomposition group on the right hand side of the above equation. Nevertheless, we will write the rest of this section in terms of general groups to avoid keeping track of multiple indices, and keep the notation as simple as possible. For the relevant group theory background, please see Section 3.3, and for a more elementary presentation, please refer to Section 2.

Notation 4.1. *For a prime p, a finite abelian group G, $s \in G$, and function $f : \mathbb{F}_p \to \widehat{G}$, we define*

$$|\varphi_G^{f,s}\rangle := \frac{1}{\sqrt{p}} \sum_{x \in \mathbb{F}_p} \chi_s(f(x))|x\rangle.$$

Here χ_s is a character of $\widehat{G}$. We call $\left(f, |\varphi_G^{f,s}\rangle\right)$ a p-ary Fourier sample. If $p = 2$, we will ignore global phase, and also call $\left(f(1) - f(0), \frac{|0\rangle + \chi_s(f(1)-f(0))|1\rangle}{\sqrt{2}}\right)$ a binary Fourier sample.

When $G = \mathbb{Z}_q^n$, the character χ_s can be chosen to be $\chi_{\mathbf{s}}(\mathbf{a}) = \omega_q^{\langle \mathbf{s}, \mathbf{a} \rangle}$ for every $\mathbf{a} \in \mathbb{Z}_q^n$.

4.1 Preparing Fourier samples

Lemma 4.2. *Let G be a finite abelian group and let $s, x \in G$ be arbitrary. Applying a Fourier transform over G to the second register of the quantum state $\frac{1}{\sqrt{2}}(|0, x\rangle + |1, x+s\rangle)$ and measuring it gives a uniformly random Fourier label $a \in \widehat{G}$ as the measurement outcome and the corresponding binary Fourier sample*

$$\frac{1}{\sqrt{2}}(|0\rangle + \chi_s(a)|1\rangle), \tag{4}$$

as the residual quantum state.

Proof. Use the Fourier transform convention $|y\rangle \mapsto |G|^{-1/2} \sum_{a \in \widehat{G}} \chi_y(a)|a\rangle$. Since $\chi_{x+s}(a) = \chi_x(a)\chi_s(a)$, applying this transform to the second register gives

$$\frac{1}{\sqrt{2|G|}} \sum_{a \in \widehat{G}} \chi_x(a)\big(|0\rangle + \chi_s(a)|1\rangle\big)|a\rangle.$$

Thus every label is measured with probability $1/|G|$. After measurement, the factor $\chi_x(a)$ is a global phase of the qubit and can be ignored, leaving the state in (4). □

Lemma 4.3. *Let p be a prime, H be a finite abelian group, and fix $s \in H$. There is a quantum procedure that takes $O(p)$ binary Fourier samples of the form $\left(a, \frac{|0\rangle + \chi_s(a)|1\rangle}{\sqrt{2}}\right)$ and outputs a p-ary Fourier sample $\left(f, |\varphi_H^{f,s}\rangle\right)$, for some $f : \mathbb{F}_p \to \widehat{H}$, $f(0) = 0$. If the binary Fourier labels are i.i.d. uniformly random, f is a uniformly random function. The procedure runs in time polynomial in p and the group-operation cost, and succeeds with probability $\Omega(1)$.*

Proof. Collect ℓ binary Fourier samples, with labels $a_1, \ldots, a_\ell$. Their joint state is

$$2^{-\ell/2} \sum_{\mathbf{b} \in \{0,1\}^\ell} \chi_s\left(\sum_{i=1}^{\ell} b_i a_i\right)|\mathbf{b}\rangle.$$

We will partition $\{0,1\}^\ell$ into components of size $\ell + 1$, measure the component label, and retain p strings from that component.

It is tempting to take $\ell = \lceil \log_2 p \rceil$ and take a single component corresponding to the binary encodings of $0, \ldots, p-1$. This succeeds with probability $p/2^\ell \geq 1/2$, but the resulting values of f are sums of only ℓ random input labels and therefore cannot be independently uniformly random. Instead, we will take $\ell = 2^{\lceil \log_2 p \rceil} - 1$, and our partition will be such that each component consists of a "central" bit string $\mathbf{c}$ and all the bit strings of Hamming distance 1 from it, i.e., $\mathbf{c} + \mathbb{1}_i$, for $i \in [\ell]$. Relative to the center's phase, the labels on these ℓ neighbors are independently uniform.

Our centers will be the vectors in the right kernel of the matrix $\mathbf{M} \in \mathbb{F}_2^{\lceil \log_2 p \rceil \times \ell}$ whose columns are all the nonzero vectors of length $\lceil \log_2 p \rceil$. The image of every other string is nonzero and therefore matches exactly one column; flipping the bit corresponding to that column changes the result to zero. Thus every string is either a center or has Hamming distance 1 from a unique center. Therefore, we can collapse the state to a single component by measuring the nearest center $\mathbf{c}$, and then we can relabel $\mathbf{c} \mapsto 0$ and $\mathbf{c} + \mathbb{1}_i \mapsto i$. The resulting state is a uniform superposition over $\ell + 1$ branches, which we then project to labels in $\mathbb{F}_p$. This succeeds with probability $p/(\ell+1) \geq 1/2$, resulting in the state, up to global phase,

$\frac{1}{\sqrt{p}} \sum_{x \in \mathbb{F}_p} \chi_s(f(x))|x\rangle$, where $f(x) = (-1)^{c_x} a_x$ for $x \neq 0$, and $f(0) = 0$. Since the a_i's are uniformly random, we have that f is uniformly random among functions with $f(0) = 0$. The success probability does not depend on the input labels or s: every branch has the same magnitude. Thus failed attempts can be discarded and replaced with fresh inputs, and the expected number of binary samples per output is $\ell(\ell+1)/p < 4p$. □

Lemma 4.4. *Let p be a prime, H be a finite abelian group, and fix $s \in H$. There is a quantum procedure that takes a p-ary Fourier sample $\left(f, |\varphi_H^{f,s}\rangle\right)$ and outputs a binary Fourier sample $\left(a, \frac{|0\rangle + \chi_s(a)|1\rangle}{\sqrt{2}}\right)$. If the function f is uniformly random conditioned on $f(0) = 0$, then a is uniformly random. The procedure runs in time polynomial in p and the group-operation cost, and succeeds with probability $\geq 2/3$.*

Proof. Partition $\mathbb{F}_p$ into fixed pairs, for instance $\{(0,1),(2,3),\ldots\}$, leaving one index unpaired if p is odd. We measure the pair index to collapse the state to be supported only on a single pair. (We declare failure if the outcome is the unpaired index.) This succeeds with probability 1 for $p = 2$ and $(p-1)/p \geq 2/3$ for odd p, independently of f and s. If the measured pair is $\{x, y\}$, we can relabel $x \mapsto 0, y \mapsto 1$. After ignoring the common phase $\chi_s(f(x))$, the normalized state is $\frac{|0\rangle + \chi_s(f(y)-f(x))|1\rangle}{\sqrt{2}}$. The procedure outputs the classical label $a = f(y) - f(x)$; and when f is uniform (subject to $f(0) = 0$), $f(y) - f(x)$ is uniform as well. □

4.2 A state combination procedure

Lemma 4.5. *Let H be a finite abelian group. Given ℓ p-ary Fourier samples with fixed secret $s \in H$ and arbitrary known labels $f_1, \ldots, f_\ell : \mathbb{F}_p \to \widehat{H}$, and a nonzero vector $\mathbf{u} \in \mathbb{F}_p^\ell$, there is a quantum procedure that measures a uniformly random coset of $\langle \mathbf{u} \rangle \leq \mathbb{F}_p^\ell$ and outputs a p-ary Fourier sample with label*

$$g(t) = \sum_{i=1}^{\ell} \big(f_i(z_i + tu_i) - f_i(z_i)\big), \tag{5}$$

along with a representative $\mathbf{z}$ of the measured coset. The representative is chosen by any fixed efficient rule that depends only on $\mathbf{u}$ and the coset. The procedure runs in time $\mathrm{poly}(\ell, p, \log|H|)$.

Proof. Given the joint input state $\frac{1}{p^{\ell/2}} \sum_{\mathbf{x} \in \mathbb{F}_p^\ell} \chi_s\left(\sum_{i=1}^{\ell} f_i(x_i)\right) |\mathbf{x}\rangle$, we coherently compute and measure which coset of $\langle \mathbf{u} \rangle$ contains the index vector $\mathbf{x}$, collapsing to a superposition on that coset. Each coset contains p orthogonal basis vectors of magnitude $p^{-\ell/2}$, so each coset occurs with probability $p^{1-\ell}$. Every vector in the measured coset is of the form $\mathbf{z} + t\mathbf{u}$ for a unique t, and the remaining state is

$$\frac{1}{\sqrt{p}} \sum_{t \in \mathbb{F}_p} \chi_s\left(\sum_{i=1}^{\ell} f_i(z_i + tu_i)\right) |\mathbf{z} + t\mathbf{u}\rangle.$$

We then reversibly relabel $|\mathbf{z} + t\mathbf{u}\rangle \mapsto |t\rangle$, so that the relative phase of $|t\rangle$ with respect to $|0\rangle$ is $\chi_s(g(t))$ in (5), with $g(0) = 0$. Computing the coset label, relabeling its p vectors, and computing the truth table of g take time polynomial in ℓ, p, and $\log|H|$. □

Lemma 4.6. *Let $f_1, \ldots, f_\ell, g$ be functions as in Lemma 4.5. If $f_1, \ldots, f_\ell$ are i.i.d. uniformly random functions that vanish on 0, and $\mathbf{u}$ is chosen independently, then g is a uniformly random function that vanishes on 0, even conditioned on $\mathbf{u}$ and $\mathbf{z}$.*

Proof. It is clear that $g(0) = 0$. Fix $\mathbf{u}$ and $\mathbf{z}$. The choice of $\mathbf{u}$ is independent of the input functions, and by Lemma 4.5, $\mathbf{z}$ is uniform for every fixed choice of those functions. Thus the posterior on f_i remains independent and uniform conditioned on $\mathbf{u}, \mathbf{z}$. Choose an index j with $u_j \neq 0$, and fix the other input functions. The contribution of f_j to g is the function $t \mapsto f_j(z_j + tu_j) - f_j(z_j)$, which vanishes at 0.

As t ranges over $\mathbb{F}_p$, the argument $z_j + tu_j$ ranges over all of $\mathbb{F}_p$, and therefore this contribution is uniform. The other functions $f_i, i \neq j$ contribute a fixed function that also vanishes at 0, and this preserves uniformity. □

Lemma 4.7. *Let $f_1, \ldots, f_\ell, g$ be functions as in Lemma 4.5, with $f_i(0) = 0$. Let $A \leq \widehat{H}$, and choose canonical representatives in $\widehat{H}$ for each coset of A, with 0 representing A. Suppose that the vector $\mathbf{u}$ in Lemma 4.5 depends only on the $[f_i]_A$ functions. Suppose further that $f_i - [f_i]_A$ are i.i.d. uniformly random A-valued functions that vanish at 0, even conditioned on the entire collection $[f_1]_A, \ldots, [f_\ell]_A$. Then, $g - [g]_A$ is a uniformly random A-valued function that vanishes at 0, even conditioned on $[g]_A$ and $\mathbf{z}$. Here $[f_i]_A$ and $[g]_A$ apply the chosen representatives pointwise.*

Proof. Fix the representative functions $[f_1]_A, \ldots, [f_\ell]_A$. This fixes $\mathbf{u}$, while the functions $f_i - [f_i]_A$ remain independent and uniform by assumption. The coset measurement of Lemma 4.5 is uniform even when all the input functions are fixed. We may therefore also condition on its outcome $\mathbf{z}$ without changing the distribution of these A-valued functions. Substituting $f_i = [f_i]_A + (f_i - [f_i]_A)$ into (5), we have

$$\begin{aligned} g(t) - [g(t)]_A = &\sum_{i=1}^{\ell} \big((f_i - [f_i]_A)(z_i + tu_i) - (f_i - [f_i]_A)(z_i)\big) \\ &+ \sum_{i=1}^{\ell} \big([f_i(z_i + tu_i)]_A - [f_i(z_i)]_A\big) - [g(t)]_A. \end{aligned}$$

The terms involving $[f_i]_A$ determine $g + A$, and hence determine $[g]_A$. Applying Lemma 4.6 to the A-valued functions $f_i - [f_i]_A$ shows that the first sum is uniform among A-valued functions vanishing at 0. The remaining terms form a fixed A-valued function: the second sum and $[g(t)]_A$ represent the same coset, and both vanish at $t = 0$. Adding this fixed function to the first sum preserves uniformity. □

4.3 Modulus reduction

Whenever $\widehat{H}/A \cong \mathbb{F}_p^r$ below, we fix such an isomorphism and use it to view $a + A$ as a vector in $\mathbb{F}_p^r$.

Lemma 4.8. *Let $f_1, \ldots, f_\ell, g : \mathbb{F}_p \to \widehat{H}$ and $\mathbf{u}$ be as in Lemma 4.5. Suppose that $\widehat{H}/A \cong \mathbb{F}_p^r$ for some $A \leq \widehat{H}$, and suppose $f_i(x) + A = \sum_{j=1}^{d} \binom{x}{j} \mathbf{a}_i^{(j)}$ for some degree $1 \leq d < p$ and coefficients $\mathbf{a}_i^{(j)} \in \mathbb{F}_p^r$. If $\mathbf{u}$ satisfies*

$$\sum_{i=1}^{\ell} u_i^d \mathbf{a}_i^{(d)} = 0, \tag{6}$$

then $g + A$ has degree at most $d - 1$. That is, there exist $\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(d-1)} \in \mathbb{F}_p^r$ such that $g(x) + A = \sum_{j=1}^{d-1} \binom{x}{j} \mathbf{b}^{(j)}$.

Proof. By (5), in $\mathbb{F}_p^r$ we have

$$g(t) + A = \sum_{i=1}^{\ell} \sum_{j=1}^{d} \left(\binom{z_i + tu_i}{j} - \binom{z_i}{j} \right) \mathbf{a}_i^{(j)}.$$

By Lemma 3.6, the coefficient of $\binom{t}{d}$ is $\sum_i u_i^d \mathbf{a}_i^{(d)} = 0$ by (6). Thus $g(t) + A$ has degree at most $d-1$. Also $g(0) + A = 0$, so its constant term vanishes. For $d = 1$, this gives $g(t) + A = 0$. □

Lemma 4.9. *Consider the setting of Lemma 4.8. Suppose that conditioned on degree-d coefficients $\mathbf{a}_1^{(d)}, \dots, \mathbf{a}_\ell^{(d)}$, the lower degree coefficients $\{\mathbf{a}_i^{(j)}\}_{i \in [\ell], j \in [d-1]}$ are jointly i.i.d. uniformly randomly distributed in $\mathbb{F}_p^r$. Suppose further that $\mathbf{u}$ is a (possibly randomized) function of the degree-d coefficients $\mathbf{a}_1^{(d)}, \dots, \mathbf{a}_\ell^{(d)}$, and independent of the lower degree coefficients. Then, the output coefficients $\mathbf{b}^{(1)}, \dots, \mathbf{b}^{(d-1)}$ are jointly i.i.d. uniform, even conditioned on $\mathbf{a}_1^{(d)}, \dots, \mathbf{a}_\ell^{(d)}$.*

Proof. The statement is vacuous for $d = 1$, so suppose $d \geq 2$. Fix the degree-d coefficients and the independent coins used to choose $\mathbf{u}$. This fixes a nonzero $\mathbf{u}$. We may also fix the measured coset representative $\mathbf{z}$: by Lemma 4.5, its distribution is uniform for every fixed collection of input tables, so this conditioning does not change the distribution of their lower coefficients.

Choose i with $u_i \neq 0$ and fix the lower coefficients of every other input. For $j < d$, the contribution of $\mathbf{a}_i^{(j)}$ to $g(t) + A$ is

$$\left(\binom{z_i + tu_i}{j} - \binom{z_i}{j} \right) \mathbf{a}_i^{(j)}.$$

By Lemma 3.6, its coefficient of $\binom{t}{j}$ is $u_i^j \mathbf{a}_i^{(j)}$, and it has no terms of higher degree. The fixed degree-d terms cancel at degree d by (6). Thus $\mathbf{b}^{(j)}$ depends on $\mathbf{a}_i^{(j)}, \dots, \mathbf{a}_i^{(d-1)}$, but not on any lower coefficient of the ith input.

At degree $d-1$, every contribution to $\mathbf{b}^{(d-1)}$ except $u_i^{d-1} \mathbf{a}_i^{(d-1)}$ is fixed. Since $u_i^{d-1} \neq 0$, this output coefficient is uniform, and fixing it also fixes $\mathbf{a}_i^{(d-1)}$. Now suppose the output coefficients of degrees $j+1, \dots, d-1$ have been fixed. Working downward in the same way, they fix $\mathbf{a}_i^{(j+1)}, \dots, \mathbf{a}_i^{(d-1)}$. The only unfixed contribution to $\mathbf{b}^{(j)}$ is then $u_i^j \mathbf{a}_i^{(j)}$. Since $\mathbf{a}_i^{(j)}$ is still independent and uniform and $u_i^j \neq 0$, $\mathbf{b}^{(j)}$ is uniform even conditioned on all higher output coefficients. Continuing to $j = 1$ proves joint independence and uniformity. □

Lemma 4.10 (Degree-relation routine)**.** *Let p be prime, $r \geq 1$, and $1 \leq d < p$, and put $g = \gcd(d, p-1)$. Set*

$$\ell(p, r, d) = (r+1)^{g \lceil \log_2(g+1) \rceil} g^{\frac{g(g-1)}{2} \lceil \log_2(g+1) \rceil}.$$

The deterministic Ivanyos–Santha algorithm, given any $\ell = \ell(p, r, d)$ vectors $\mathbf{a}_1, \dots, \mathbf{a}_\ell \in \mathbb{F}_p^r$, runs in time at most $T(p, r, d) = \mathrm{poly}(\ell(p, r, d), p)$ and finds a nonzero $\mathbf{u} \in \mathbb{F}_p^\ell$ satisfying

$$\sum_{i=1}^{\ell} u_i^d \mathbf{a}_i = \mathbf{0}.$$

In particular, $\ell(p, r, 1) = r + 1$.

Proof. Apply Theorem 3.4 with $n = r$. The chosen number of vectors meets its hypothesis, giving the required relation and running time. When $d = 1$, we have $g = 1$ and the formula gives $r + 1$ vectors. In the degree-reduction procedure, the algorithm inspects only the supplied degree-d coefficient vectors. □

Below, $\ell(p, r, d)$ and $T(p, r, d)$ refer to the routine in Lemma 4.10.

Lemma 4.11. *Let p be prime, let $r \geq 1$ be an integer, H be a finite abelian group, and let $A \leq \widehat{H}$ with $\widehat{H}/A \cong \mathbb{F}_p^r$. There is a Las Vegas quantum procedure that takes as input p-ary Fourier samples $\left(f_i, |\varphi_H^{f_i,s}\rangle\right)$ over H with fixed secret s and i.i.d. uniformly random functions $f_i : \mathbb{F}_p \to \widehat{H}$ that vanish at 0, and outputs a p-ary Fourier sample $\left(g, |\varphi_H^{g,s}\rangle\right)$ with*

$$g(t) + A = t\mathbf{a}$$

for all $t \in \mathbb{F}_p$, where $\mathbf{a} \sim \mathbb{F}_p^r$ is uniformly random. Further, we have that $g(0) = 0$, and conditioned on $g + A$, the values of $g(t)$ for $t \neq 0$ are independently uniform in the corresponding coset of A. The expected number of input samples and time required by the procedure are at most m and T' respectively, where

$$m = \prod_{d=2}^{p-1} \ell(p, r, d), \qquad \text{and} \qquad T' = O\left(p \prod_{d=2}^{p-1} \left(T(p, r, d) + \mathrm{poly}(\ell(p, r, d), p, r, \log |H|)\right)\right).$$

Proof. A degree-d pass takes samples whose quotient coefficients of degrees $d+1, \ldots, p-1$ have already been cleared, and clears their degree-d coefficients. Begin with a supply of fresh original input samples. For $d = p-1, p-2, \ldots, 2$, the degree-d pass does the following:

1. Draw $\ell := \ell(p, r, d)$ independent samples $(f_i, |\varphi_H^{f_i,s}\rangle), i \in [\ell]$ from the previous pass, and compute the degree-d coefficient $\mathbf{a}_i^{(d)}$ of the functions f_i modulo A:

$$\mathbf{a}_i^{(d)} = \sum_{x=0}^{d} (-1)^{d-x} \binom{d}{x} \big(f_i(x) + A\big).$$

2. Run the degree-d relation routine of Lemma 4.10 on these degree-d coefficients $\mathbf{a}_1^{(d)}, \ldots, \mathbf{a}_\ell^{(d)}$ to get $\mathbf{u}$ such that (6) holds.

3. Apply Lemma 4.5 to the block using $\mathbf{u}$, and provide its output as one sample from the new supply.

Return one sample from the final degree pass. For each $t \neq 0$, the pair $\big(f_i(t) + A,\ f_i(t) - [f_i(t)]_A\big)$ is uniform in $\mathbb{F}_p^r \times A$. These pairs are independent across i and t. Thus the A-valued parts are uniform even after the functions f_i are fixed modulo A, and Lemma 3.5 gives

$$f_i(t) + A = \sum_{j=1}^{p-1} \binom{t}{j} \mathbf{a}_i^{(j)},$$

where the $\mathbf{a}_i^{(j)}$ are independent uniform vectors in $\mathbb{F}_p^r$. Suppose the samples entering a degree-d pass have

$$f_i(t) + A = \sum_{j=1}^{d} \binom{t}{j} \mathbf{a}_i^{(j)},$$

where the $\mathbf{a}_i^{(j)}$ are independent uniform. The relation returned by the deterministic routine and Lemmas 4.5 and 4.8 give an output sample with

$$g(t) + A = \sum_{j=1}^{d-1} \binom{t}{j} \mathbf{b}^{(j)}.$$

The lower output coefficients $\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(d-1)}$ remain uniform, by Lemma 4.9. Conditional on $f_i + A$, the functions $f_i - [f_i]_A$ remain independent uniform. Since $\mathbf{u}$ depends only on the functions $f_i + A$, by Lemma 4.7, we have that $g - [g]_A$ is uniform even after $g + A$ is fixed. Thus the next pass receives the same uniformity properties.

After the final pass, $g(t) + A = t\mathbf{a}$ with uniform $\mathbf{a} \in \mathbb{F}_p^r$, and conditional on $g + A$, the values $g(t) - [g(t)]_A$ for $t \neq 0$ are independent uniform in A, as required. For $p = 2$, the returned input sample already has these properties.

Each degree-d pass consumes exactly $\ell(p, r, d)$ outputs from the preceding pass, because its relation routine always returns a valid direction. Multiplying these costs gives m original input samples per output. The additional time per pass is at most $T(p, r, d)$ plus a polynomial in $\ell(p, r, d)$, p, r, and $\log |H|$ for coefficient computation and state combination. Choose the polynomial term to dominate $\ell(p, r, d)$. The time recurrence then has fewer than p degree levels, and summing their costs gives the displayed bound on T'. For $p = 2$, both products are empty, so the returned input also meets the bounds. □

Lemma 4.12. *Let p be a prime, let $r \geq 1$ be an integer, let H be a finite abelian group with $A \leq \widehat{H}$ such that $\widehat{H}/A \cong \mathbb{F}_p^r$. Let $K := A^\perp \cong \mathbb{F}_p^r$. Then, there exists a Las Vegas quantum algorithm that takes as input p-ary Fourier samples $\left(f_i, |\varphi_H^{f_i,s}\rangle\right)$ over H with fixed secret $s \in H$ and i.i.d. uniformly random functions $f_i : \mathbb{F}_p \to \widehat{H}$ that vanish at 0, and outputs a p-ary Fourier sample $\left(g, |\varphi_{H/K}^{g,s+K}\rangle\right)$ over H/K. Further, $g : \mathbb{F}_p \to A$ is a uniformly random function that vanishes at 0. This procedure uses at most m input samples in expectation, and runs in time at most T' in expectation, where*

$$m = \prod_{d=1}^{p-1} \ell(p, r, d), \qquad \text{and} \qquad T' = O\left(p \prod_{d=1}^{p-1} \left(T(p, r, d) + \mathrm{poly}(\ell(p, r, d), p, r, \log |H|)\right)\right).$$

Proof. Use Lemma 4.11 on disjoint fresh inputs to obtain independent samples $\left(f_i, |\varphi_H^{f_i,s}\rangle\right)$ for which $f_i(t) + A = t\mathbf{a}_i$, with the $\mathbf{a}_i$ independent and uniform in $\mathbb{F}_p^r$. Collect $\ell(p, r, 1)$ such samples and run the degree-1 routine from Lemma 4.10 on these vectors. Combining the states using Lemma 4.5 gives a p-ary Fourier sample $\left(g, |\varphi_H^{g,s}\rangle\right)$ such that

$$g(t) + A = t \sum_{i=1}^{\ell} u_i \mathbf{a}_i = 0.$$

Since $g(t) + A = A$ and 0 represents A, we have $[g]_A = 0$ and hence $g - [g]_A = g$. By Lemma 4.7, we have that g is uniformly random, conditioned on $g(0) = 0$. For every $a \in A$ and $k \in K = A^\perp$, we have $\chi_{s+k}(a) = \chi_s(a)\chi_k(a) = \chi_s(a)$. Under $A \cong \widehat{H/K}$, this gives $|\varphi_H^{g,s}\rangle = |\varphi_{H/K}^{g,s+K}\rangle$.

A degree-1 pass uses $\ell(p, r, 1)$ outputs of Lemma 4.11, giving the stated sample bound m. The relation routine and final state combination add $T(p, r, 1) + \mathrm{poly}(\ell(p, r, 1), p, r, \log |H|)$ time. Applying the same time recurrence as in Lemma 4.11 and absorbing the fewer than p degree levels gives the stated bound T'. □

4.4 Terminal recovery

Lemma 4.13. *Let p be prime, let $r \geq 1$ be an integer, and let H be a finite abelian group with fixed coordinates $\widehat{H} \cong \mathbb{F}_p^r$. For $a \in \widehat{H}$, write $\mathbf{a} \in \mathbb{F}_p^r$ for its coordinates; the unknown $s \in H$ is represented by $\mathbf{s} \in \mathbb{F}_p^r$ through $\chi_s(a) = \omega_p^{\langle \mathbf{a}, \mathbf{s} \rangle}$. There is a Las Vegas quantum procedure that takes independent p-ary Fourier samples $\left(f_i, |\varphi_H^{f_i,s}\rangle\right)$ over H, where the functions $f_i : \mathbb{F}_p \to \widehat{H}$ are i.i.d. uniform and vanish at 0, and returns $\mathbf{s}$ with probability 1. Its expected number of input samples is at most m, and its expected running time is at most T', where*

$$m = (r+2)\prod_{d=2}^{p-1} \ell(p,r,d), \qquad \text{and} \qquad T' = O\left(m \ \mathrm{poly}\left(p, r, \log|H|, \sum_{d=2}^{p-1}\big(\ell(p,r,d) + T(p,r,d)\big)\right)\right).$$

Proof. Apply Lemma 4.11 with $A = \{0\}$ to fresh, disjoint inputs. Each call returns a sample with $g(t) = t\mathbf{a}$, where $\mathbf{a}$ is uniform in $\mathbb{F}_p^r$. Its state is

$$|\varphi_H^{g,s}\rangle = \frac{1}{\sqrt{p}} \sum_{t \in \mathbb{F}_p} \omega_p^{t\langle \mathbf{a}, \mathbf{s} \rangle} |t\rangle.$$

An inverse Fourier transform over $\mathbb{F}_p$ followed by a measurement gives $\langle \mathbf{a}, \mathbf{s} \rangle$ exactly. Calls on disjoint inputs therefore give independent uniform vectors $\mathbf{a}$ and their corresponding linear equations in $\mathbf{s}$. Continue until the obtained vectors $\mathbf{a}$ span $\mathbb{F}_p^r$, then solve the equations by Gaussian elimination.

When their current span has dimension j, a fresh uniform vector lies outside it with probability $1 - p^{j-r}$. Thus the expected number of calls is

$$\sum_{j=0}^{r-1} \frac{1}{1-p^{j-r}} = r + \sum_{j=1}^{r} \frac{1}{p^j - 1} < r + 2,$$

since $1/(p^j - 1) \leq 2^{1-j}$ and $\sum_{j=1}^{r} 2^{1-j} < 2$, giving the desired upper bound on the expected sample complexity.

Since each original sample is used in at most one attempt per degree and each final equation uses at least one original sample, the total work is at most the number of original samples times a polynomial in $p, r, \log|H|$ and $\sum_{d=2}^{p-1}(\ell(p,r,d) + T(p,r,d))$, giving the stated expected time bound. □

Lemma 4.14. *Let p be prime, let H be a finite abelian group with fixed secret $s \in H$, and let $A \leq \widehat{H}$. Let $t \in H$ be such that χ_t is a known, efficiently evaluatable extension of $\chi_s|_A$. Define $K := A^{\perp} \leq H$, so that $\widehat{K} \cong \widehat{H}/A$. Then a deterministic quantum procedure takes as input a p-ary Fourier sample $\left(f, |\varphi_H^{f,s}\rangle\right)$ with $f(0) = 0$ and $\chi_s|_A$, multiplies the phase of $|x\rangle$ by $\chi_t(f(x))^{-1}$, and outputs a p-ary Fourier sample $\left(g, |\varphi_K^{g,r}\rangle\right)$ over K, where $g : \mathbb{F}_p \to \widehat{H}/A$,*

$$r = s - t \in K, \qquad \text{and} \qquad g(x) = f(x) + A.$$

If f is a uniformly random function such that $f(0) = 0$, so is g.

Proof. Since χ_t extends $\chi_s|_A$, for every $h \in A$ we have $\chi_{s-t}(h) = \chi_s(h)\chi_t(h)^{-1} = 1$. Hence $r = s - t$ belongs to $K = A^{\perp}$. In particular, χ_r is trivial on A, so its value at $f(x)$ depends only on the coset

$g(x) = f(x) + A$. The prescribed phase correction therefore changes the state to

$$\frac{1}{\sqrt{p}} \sum_{x \in \mathbb{F}_p} \chi_s(f(x))\chi_t(f(x))^{-1}|x\rangle = \frac{1}{\sqrt{p}} \sum_{x \in \mathbb{F}_p} \chi_r(g(x))|x\rangle = |\varphi_K^{g,r}\rangle,$$

using $\widehat{H}/A \cong \widehat{K}$. Since f is uniform subject to $f(0) = 0$, the values of $g(x) = f(x) + A$ for $x \neq 0$ are independent and uniform in $\widehat{H}/A$. □

4.5 Putting everything together

We are now ready to describe the main algorithm.

Algorithm 4.15. *As input, we are given black-box access to a finite abelian group G, and independent coset states $\frac{1}{\sqrt{2}}\big(|0, x\rangle + |1, x + s\rangle\big)$, where each x is independent and uniform in G and $s \in G$ is fixed and unknown. Use the Ivanyos–Santha degree-relation routine from Lemma 4.10. Then, we describe an algorithm to find $s \in G$ as follows:*

1. *Use the Cheung–Mosca algorithm [CM01] to compute an explicit cyclic decomposition of G, and fix a chain*
$$\widehat{G} = A_0 \supsetneq A_1 \supsetneq \cdots \supsetneq A_k = \{0\} \tag{7}$$
such that $A_{i-1}/A_i \cong \mathbb{F}_{p_i}^{r_i}$ for all $i \in [k]$, where each p_i is prime and $r_i \geq 1$.

2. *Initialize χ to the unique character on $A_k = \{0\}$, that is, $\chi(0) = 1$.*

3. *Whenever a fresh binary Fourier sample is needed, apply Lemma 4.2 to a new coset state to obtain the binary Fourier sample $\Big(a, \frac{1}{\sqrt{2}}\left(|0\rangle + \chi_s(a)|1\rangle\right)\Big)$, where a is uniform in $\widehat{G}$.*

4. *For $i = k, k-1, \ldots, 1$, with $\chi = \chi_s|_{A_i}$ at the start of the stage:*

   (a) *Use the explicit cyclic decomposition to find $t \in G$ such that $\chi_t|_{A_i} = \chi$. Apply Lemma 4.14 to each fresh binary Fourier sample with $p = 2$, $H = G$, $A = A_i$, $K = A_i^\perp$, and the extension χ_t. For a sample with label a, the lemma's input function has $f(0) = 0$ and $f(1) = a$: multiply the phase of $|1\rangle$ by $\chi_t(a)^{-1}$ and replace its label by $a + A_i$. The lemma's r is $s - t \in A_i^\perp$, and the corrected samples have independent uniform labels in $\widehat{G}/A_i$.*

   (b) *For $j = 1, \ldots, i-1$, turn binary samples with independent uniform labels in A_{j-1}/A_i into binary samples with independent uniform labels in A_j/A_i as follows:*

      i. *Use Lemma 4.3 with $p = p_j$, $H = A_i^\perp/A_{j-1}^\perp$, and secret $(s-t) + A_{j-1}^\perp$ to convert the current binary samples into p_j-ary Fourier samples. Here $\widehat{H} \cong A_{j-1}/A_i$.*

      ii. *Apply Lemma 4.12 with $p = p_j$, $r = r_j$, $H = A_i^\perp/A_{j-1}^\perp$, $A = A_j/A_i$, $K = A^\perp = A_j^\perp/A_{j-1}^\perp$, and secret $(s-t) + A_{j-1}^\perp$. Indeed, $\widehat{H}/A \cong A_{j-1}/A_j \cong \mathbb{F}_{p_j}^{r_j}$. Use the Ivanyos–Santha routine from Lemma 4.10 for degrees $1, \ldots, p_j - 1$; at degree one it takes $r_j + 1$ coefficient vectors. The output p_j-ary samples have independent uniform label functions into A_j/A_i and secret $(s-t) + A_j^\perp$ over $H/K \cong A_i^\perp/A_j^\perp$.*

      iii. *Use Lemma 4.4 with $p = p_j$, $H = A_i^\perp/A_j^\perp$, and secret $(s-t) + A_j^\perp$ to convert the outputs back to binary samples with independent uniform labels in $\widehat{H} \cong A_j/A_i$.*

(c) *Use Lemma 4.3 with $p = p_i$, $H = A_i^\perp/A_{i-1}^\perp$, and secret $(s-t) + A_{i-1}^\perp$ to convert independent binary samples with uniform labels in $\widehat{H} \cong A_{i-1}/A_i$ into p_i-ary Fourier samples. When $i = 1$, these binary samples come directly from the character correction in part (a).*

(d) *Apply Lemma 4.13 with $p = p_i$, $r = r_i$, $H = A_i^\perp/A_{i-1}^\perp$, secret $(s-t) + A_{i-1}^\perp$, and coordinates $\widehat{H} \cong A_{i-1}/A_i \cong \mathbb{F}_{p_i}^{r_i}$. Its output determines the character of A_{i-1}/A_i induced by $s-t$.*

(e) *Extend χ to A_{i-1} by setting*

$$\chi(a) = \chi_t(a)\chi_{s-t}(a + A_i) \qquad \text{for every } a \in A_{i-1},$$

where the second factor is the character recovered in part (d).

5. *Identify s from $\chi = \chi_s$ on $A_0 = \widehat{G}$ using the explicit cyclic coordinates, and output s.*

Theorem 4.16. *Let G be a finite abelian group. Suppose $\widehat{G}$ has a chain with effective coordinates*

$$\widehat{G} = A_0 \supsetneq A_1 \supsetneq \cdots \supsetneq A_k = \{0\}, \qquad A_{i-1}/A_i \cong \mathbb{F}_{p_i}^{r_i} \quad (i \in [k]),$$

where each p_i is prime and $r_i \geq 1$. For every i and $2 \leq d < p_i$, use the deterministic degree-d relation routine with parameters $\ell(p_i, r_i, d)$ and $T(p_i, r_i, d)$ from Lemma 4.10. Then, with probability at least $\Omega(1)$, Algorithm 4.15 solves the generalized dihedral coset problem over $\mathrm{dih}(G)$ with at most m samples and T time, where

$$m = O\left(\prod_{i=1}^{k}\left(8p_i(r_i+1)\prod_{d=2}^{p_i-1}\ell(p_i,r_i,d)\right)\right), \text{ and}$$

$$T = O\left(m \cdot \mathrm{poly}\left(\log(m|G|), \sum_{i=1}^{k}\left(p_i + \sum_{d=2}^{p_i-1}\left(\ell(p_i,r_i,d) + T(p_i,r_i,d)\right)\right)\right)\right).$$

Proof. The explicit cyclic decomposition supplies the subgroup, quotient, and character operations needed in the algorithm. Each quotient A_{i-1}/A_i has order $p_i^{r_i} \geq 2$, so the chain has $k \leq \log_2 |G|$ levels. By Lemma 4.2, Step 3 turns each fresh coset state into a binary Fourier sample with a uniform label in $\widehat{G}$.

We claim that the algorithm maintains the following invariant: at the start of stage i in the Step 4 loop, we have that $\chi = \chi_s|_{A_i}$. We prove this by induction. The base case $i = k$, when χ is initialized in Step 2, holds since the unique character on $A_k = \{0\}$ is $\chi_s|_{A_k}$.

By the induction hypothesis, suppose that at the start of stage i in Step 4, we have that $\chi = \chi_s|_{A_i}$. The explicit cyclic decomposition isomorphism allows us to find some $t \in G$ such that $\chi_t|_{A_i} = \chi$. Hence $s - t \in A_i^\perp$.

Step 4(a). Correcting for the recovered parts of s. Applying Lemma 4.14 in Step 4(a) with $p = 2$ to fresh binary samples gives independent samples with secret $s - t$ and uniform labels in $\widehat{G}/A_i$.

Step 4(b). Modulus reduction from $A_0 := \widehat{G}$ to A_{i-1}. At the start of pass j in Step 4(b), the binary labels are independent and uniform in A_{j-1}/A_i. By Lemma 4.3, they give p_j-ary samples over $A_i^\perp/A_{j-1}^\perp$ with independent uniform label functions, with at most $4p_j$ binary Fourier samples in expectation per output.

In Step 4(b)(ii), for the current group $H = A_i^\perp/A_{j-1}^\perp$, the subgroup $A = A_j/A_i$ satisfies $\widehat{H}/A \cong A_{j-1}/A_j \cong \mathbb{F}_{p_j}^{r_j}$, and $K = A^\perp = A_j^\perp/A_{j-1}^\perp$. The Ivanyos–Santha routine from Lemma 4.10 finds a nonzero degree-one relation among $r_j + 1$ coefficient vectors. Thus Lemma 4.12 gives a p_j-ary sample over $H/K \cong A_i^\perp/A_j^\perp$ with secret $(s-t) + A_j^\perp$ and a uniform label function into A_j/A_i. It uses at most

$$(r_j + 1) \prod_{d=2}^{p_j-1} \ell(p_j, r_j, d)$$

input p_j-ary samples. Calls on disjoint fresh inputs give independent outputs. By Lemma 4.4, a successful conversion of an output of part (ii) gives a binary sample with a uniform label in A_j/A_i. The conversion succeeds with probability at least $2/3$, so it uses fewer than two p_j-ary samples in expectation. Both conversion success probabilities are independent of the labels, so retrying with fresh samples preserves the uniform independent output labels. Combining steps 4(b)(i)–(iii), one binary output of pass j costs fewer than $8p_j(r_j+1)\prod_{d=2}^{p_j-1} \ell(p_j, r_j, d)$ binary inputs from the preceding pass in expectation.

Step 4(c). After pass $i-1$, the binary labels are independent and uniform in A_{i-1}/A_i; when $i = 1$, this follows directly from part (a). By Lemma 4.3, fewer than $4p_i$ such binary samples in expectation give one p_i-ary sample with a uniform label function.

Step 4(d). The dual of $A_i^\perp/A_{i-1}^\perp$ is $A_{i-1}/A_i \cong \mathbb{F}_{p_i}^{r_i}$. Thus Lemma 4.13 recovers the character induced by $s-t$ on this quotient with probability 1. It uses at most

$$(r_i + 2) \prod_{d=2}^{p_i-1} \ell(p_i, r_i, d)$$

p_i-ary samples in expectation. With Step 4(c), the expected number of binary samples needed after the inner loop is less than $4p_i(r_i+2)\prod_{d=2}^{p_i-1} \ell(p_i, r_i, d)$.

Step 4(e). Since $s - t \in A_i^\perp$, its character is well defined on A_{i-1}/A_i. For every $a \in A_{i-1}$, the update in the algorithm therefore gives

$$\chi(a) = \chi_t(a)\chi_{s-t}(a + A_i) = \chi_t(a)\chi_{s-t}(a) = \chi_s(a).$$

Hence $\chi = \chi_s|_{A_{i-1}}$, as required for the next stage.

Therefore, after stage $i = 1$, we know χ_s on $A_0 = \widehat{G}$. The explicit cyclic decomposition and Lemma 3.9 allow us to find the unique $s \in G$ with this character.

Time and sample complexity. The expected sample costs multiply because each pass combines samples produced by the preceding pass, and summing over recovery stages changes the full product by only a constant factor, giving the stated bound m in expectation. The calls to [IS17] and polynomial work per sample give the corresponding expected running time T. Aborting runs that exceed sufficiently large constant multiples of these expectations gives the stated time and sample bounds with constant success probability by Markov's inequality. □

Corollary 4.17 (Finite abelian groups of smooth exponent)**.** *Let G be a finite abelian group whose exponent is B-smooth, where $B \geq 2$. Suppose $\widehat{G}$ has a chain with effective coordinates $\widehat{G} = A_0 \supsetneq \cdots \supsetneq A_k = \{0\}$ and $A_{i-1}/A_i \cong \mathbb{F}_{p_i}^{r_i}$ for every $i \in [k]$, where each p_i is prime and $r_i \geq 1$. Then, for every $0 < \varepsilon < 1$, the*

generalized dihedral coset problem over $\mathrm{dih}(G)$ *can be solved with probability at least* $1-\varepsilon$*. Both its sample complexity and running time are at most*

$$\mathrm{poly}\left(B, \log|G|, \log\frac{2}{\varepsilon}\right) B^{O(kB^{2+o(1)})} \prod_{i=1}^{k} (1+r_i)^{O(B^{1+o(1)})}.$$

Proof. Fix a quotient $\mathbb{F}_p^r$ and a degree d, and put $g = \gcd(d, p-1)$ as in Lemma 4.10. Since $g < p$ and $\lceil \log_2(g+1) \rceil = O(\log p)$, its number of input vectors satisfies

$$\ell(p, r, d) \le (r+1)^{O(g \log p)} p^{O(g^2 \log p)}.$$

This also holds when $g = 1$, for which $\ell(p, r, d) = r + 1$.

For each divisor g of $p-1$, at most $(p-1)/g$ degrees have $\gcd(d, p-1) = g$, since each such degree is a multiple of g. Grouping the factors by this gcd gives

$$\begin{aligned}\prod_{d=2}^{p-1} \ell(p, r, d) &\le (r+1)^{O(p \log p \, \#\{g \ge 1 : g \mid p-1\})} p^{O\left(p \log p \sum_{g \mid p-1} g\right)} \\ &\le (r+1)^{O(p^{1+o(1)})} p^{O(p^{2+o(1)})}.\end{aligned}$$

For the second inequality, Lemma 3.11 gives $\#\{g \ge 1 : g \mid p-1\} = p^{o(1)}$, and each divisor is less than p, so $\sum_{g \mid p-1} g \le p \, \#\{g \ge 1 : g \mid p-1\}$. The factor $\log p$ is absorbed into $p^{o(1)}$. When $p = 2$, the product is empty and the same bound holds.

Multiplying by $8p(r+1)$ preserves the last bound, so Theorem 4.16 bounds the sample complexity by the product of these factors over the chain. Every $p_i \le B$, giving the stated separate powers of B and $1 + r_i$. The time bound has the same form: $T(p, r, d)$ is polynomial in $\ell(p, r, d)$ and p, and the theorem's additional polynomial work only changes the absolute constants in these exponents.

Repeating the capped algorithm $O(\log(2/\varepsilon))$ times and allocating total approximation error at most $\varepsilon/2$ gives the stated success probability and costs.

□

The vector DCP guarantee follows by taking the subgroup chain that removes one prime factor of q at a time.

Corollary 4.18 (Vector DCP over smooth moduli)**.** *Let* $n \ge 1$*, let* $q > 1$ *be* B*-smooth, and let* $0 < \varepsilon < 1$*. A quantum algorithm recovers every* $\mathbf{s} \in \mathbb{Z}_q^n$ *from independent vector DCP samples with probability at least* $1-\varepsilon$*. Both its sample complexity and running time are at most*

$$n^{O(B^{1+o(1)} \log q)} B^{O(B^{2+o(1)} \log q)} \, \mathrm{poly}\left(\log\frac{2}{\varepsilon}\right).$$

Proof. Write $q = \prod_{i=1}^{k} p_i$, with repetition, and take $A_0 = \widehat{\mathbb{Z}_q^n} \cong \mathbb{Z}_q^n$ and $A_i = (p_1 \cdots p_i) A_0$. Then $A_{i-1}/A_i \cong \mathbb{F}_{p_i}^n$, the coordinates are given by modular arithmetic, and $k \le \log_2 q$. Apply Corollary 4.17 with $r_i = n$ and $|G| = q^n$. Its product over the chain is at most $n^{O(B^{1+o(1)} \log q)} B^{O(B^{2+o(1)} \log q)}$. The polynomial overhead is absorbed into this bound, apart from the stated dependence on ε.

□

References

[BFS26] Mathias Boucher, Pierre-Alain Fouque, and Yixin Shen. Cyclotomic cosets: Hidden subgroup and quantum sieving algorithm for prime-power moduli. *arXiv preprint arXiv:2609.34996*, 2026. 4, 6, 7

[BJK+25] Shi Bai, Hansraj Jangir, Elena Kirshanova, Tran Ngo, and William Youmans. A quasi-polynomial time algorithm for the extrapolated dihedral coset problem over power-of-two moduli. *IACR Cryptology ePrint Archive, Paper 2025/1046*, 2025. 3, 6, 8, 9

[BKSW18] Zvika Brakerski, Elena Kirshanova, Damien Stehlé, and Weiqiang Wen. Learning with errors and extrapolated dihedral cosets. In *IACR international workshop on public key cryptography*, pages 702–727. Springer, 2018. 3, 8

[BNP18] Xavier Bonnetain and María Naya-Plasencia. Hidden shift quantum cryptanalysis and implications. In *Advances in Cryptology—ASIACRYPT 2018, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 560–592. Springer, 2018. 6, 9

[BS20] Xavier Bonnetain and André Schrottenloher. Quantum security analysis of CSIDH. In *Advances in Cryptology—EUROCRYPT 2020, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 493–522. Springer, 2020. 3

[CDEL21] Wouter Castryck, Ann Dooms, Carlo Emerencia, and Alexander Lemmens. A fusion algorithm for solving the hidden shift problem in finite abelian groups. In *Post-Quantum Cryptography—PQCrypto 2021*, volume 12841 of *Lecture Notes in Computer Science*, pages 133–153. Springer, 2021. 6

[CJS14] Andrew M. Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014. 3

[CLZ22] Yilei Chen, Qipeng Liu, and Mark Zhandry. Quantum algorithms for variants of average-case lattice problems via filtering. In *Advances in Cryptology – EUROCRYPT 2022, Part III*, volume 13277 of *Lecture Notes in Computer Science*, pages 372–401. Springer, 2022. 6

[CM01] Kevin K. H. Cheung and Michele Mosca. Decomposing finite abelian groups. *Quantum Information and Computation*, 1(3):26–32, 2001. 14, 22

[Csá21] Gergely Csáji. A new quantum algorithm for the hidden shift problem in $\mathbb{Z}_{2^t}^n$. *arXiv preprint arXiv:2102.04171*, 2021. 6, 9

[EHK04] Mark Ettinger, Peter Høyer, and Emanuel Knill. The quantum query complexity of the hidden subgroup problem is polynomial. *Information Processing Letters*, 91(1):43–48, 2004. 3

[FIM+14] Katalin Friedl, Gábor Ivanyos, Frédéric Magniez, Miklos Santha, and Pranab Sen. Hidden translation and translating coset in quantum computing. *SIAM Journal on Computing*, 43(1):1–24, 2014. 3, 5, 6

[II23] Muhammad Imran and Gábor Ivanyos. Zero sum subsequences and hidden subgroups. *arXiv preprint arXiv:2304.08376*, 2023. 4, 6, 7, 10, 11, 12

[IS17] Gábor Ivanyos and Miklós Santha. Solving systems of diagonal polynomial equations over finite fields. *Theoretical Computer Science*, 657:73–85, 2017. 4, 7, 8, 12, 24

[Iva08] Gábor Ivanyos. On solving systems of random linear disequations. *Quantum Information and Computation*, 8(6–7):579–594, 2008. 5

[KMO+26] Robin Kothari, Tony Metger, Ryan O'Donnell, Noah Shutty, and Kewen Wu. Exponential quantum speedup for $\mathbb{F}_3^n$-Subset-Sum? Or, rigorous classical algorithms for Binary-Error LWE, 2026. 7

[KOW26] Robin Kothari, Ryan O'Donnell, and Kewen Wu. No exponential quantum speedup for SIS^∞ anymore. *arXiv preprint arXiv:2510.07515*, 2026. 4, 7, 11, 12

[Kup05] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005. 3, 4, 9

[Kup13] Greg Kuperberg. Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. In *8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2013)*, volume 22 of *LIPIcs*, pages 20–34, 2013. 5

[Pei20] Chris Peikert. He gives C-sieves on the CSIDH. In *Advances in Cryptology—EUROCRYPT 2020, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 463–492. Springer, 2020. 3, 5

[Reg04a] Oded Regev. Quantum computation and lattice problems. *SIAM Journal on Computing*, 33(3):738–760, 2004. 3

[Reg04b] Oded Regev. A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space. *arXiv preprint quant-ph/0406151*, 2004. 3, 5

[Sim97] Daniel R Simon. On the power of quantum computation. *SIAM journal on computing*, 26(5):1474–1483, 1997. 5, 8