A Polynomial-Time Quantum Algorithm for the Dihedral Coset Problem

DANIEL R. SIMON, Amazon Web Services, USA

We present a polynomial-time quantum algorithm for the Dihedral Coset Problem (DCP). The algorithm is based on Regev's polynomial-time reduction of the Dihedral Subgroup Problem (DSP) to the modular subset sum problem [12], but uses a different technique to erase sample bits without use of a subset sum oracle. The algorithm can thus combine with Regev's [12] reduction of lattice problems to DCP, improved by Brakerski, Kirshanova, Stehlé and Wen [3], to yield polynomial-time quantum algorithms for various lattice problems, such as finding a polynomial-factor approximation to the shortest vector in an n-dimensional lattice (SVP), and the "learning with errors" problem (LWE). The algorithm can tolerate a faulty sample rate as high as $1/O(\log n)$, allowing the algorithm-reduction combination to efficiently solve, for example, SVP with a $\sqrt{n}$ polylog(n) approximation factor, or LWE instances with $\alpha = \sqrt{n}$ polylog(n).

CCS Concepts: • **Theory of computation** → **Quantum computation theory**; *Cryptographic primitives*; • **Security and privacy** → Public key (asymmetric) techniques.

Additional Key Words and Phrases: quantum algorithms, dihedral coset problem, hidden subgroup problem, lattice-based cryptography, learning with errors

ACM Reference Format:
Daniel R. Simon. 2026. A Polynomial-Time Quantum Algorithm for the Dihedral Coset Problem. 1, 1 (November 2026), 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

1.1 Lattice-Based Cryptography

The discovery of quantum algorithms capable of efficiently attacking the most popular classical public-key cryptosystems [14] has created interest in public-key cryptosystems that are resistant to such quantum attacks. One promising approach is "lattice cryptography" [10], which is based on the believed computational difficulty of various problems over lattices, such as the polynomial-factor approximate Shortest Vector Problem ($p(n)$-SVP). $p(n)$-SVP is the problem of finding a non-zero vector in an n-dimensional lattice whose length is within a polynomial $p(n)$ factor of the length of the shortest non-zero vector in the lattice. The security of several lattice-based cryptosystems, such as the Ajtai-Dwork cryptosystem [1] and "learning with errors" (LWE) [13], depends on the hardness of $p(n)$-SVP. The best previously available algorithms—classical or quantum—for solving $p(n)$-SVP, such as BKZ ([4]), require fully exponential time.

Author's Contact Information: Daniel R. Simon, Amazon Web Services, Cryptography Group, Seattle, Washington, USA, dcp-paper@amazon.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM XXXX-XXXX/2026/11-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1.2 The Dihedral Subgroup Problem

The Dihedral Subgroup Problem (DSP) [5] can be stated as follows: assume that a function f is constant on a subgroup H of a dihedral group G, as well as on each of its cosets; the problem is to identify H. Ettinger and Høyer [5] showed that it suffices to solve the problem in the case where the subgroup is of order 2. Applying the standard "sampling" method results in the following very simple formulation of the problem, due to Regev [12]: given a source of random samples of the superposition

$$\frac{1}{\sqrt{2}}|0, x\rangle + \frac{1}{\sqrt{2}}|1, x + d \pmod N\rangle$$

where $2N \approx 2^{n+1}$ is the size of the dihedral group, $d, x \in \{0...N-1\}$, d is fixed and x is chosen arbitrarily for each sample, find d.

Boneh and Lipton [2] showed that the general hidden subgroup problem for Abelian groups is in quantum polynomial time, but dihedral groups are (slightly) non-Abelian. The best previously known quantum algorithm for solving DSP is due to Kuperberg [8], and has running time $2^{O(\sqrt{\log N})} \approx 2^{O(\sqrt{n})}$. Regev [12] gives a polynomial-time quantum algorithm for DSP, but it relies on a subset sum oracle to "erase" input bits. The algorithm presented here emulates Regev's algorithm, but handles the input bits differently.

1.3 The Dihedral Coset Problem

The Dihedral Coset Problem (DCP) [12] is identical to Regev's formulation of the DSP, but with a faulty sample probability of $1/a(n)$ in the DSP samples—that is, with probability $1/a(n)$ that the sample will consist of a random bit and random value, rather than the correct superposition. Regev [12] showed a polynomial-time reduction from solving $a(n)$-SVP to solving DCP on a group of size $2N$, where $N \approx 2^{n^2}$. This quadratic dimension increase was later removed in an improved reduction from LWE presented by Brakerski, Kirshanova, Stehlé and Wen [3]. These reductions both yield an approximation factor $a(n)\sqrt{n}$ polylog(n), but at the cost of a $1/a(n)$ faulty sample probability. Kuperberg's algorithm requires error-free input, and hence produces an approximation factor at best $2^{O(\sqrt{n})}$ when combined with Regev's reduction—no better than what BKZ offers classically. Since the quantum algorithm presented here tolerates a faulty sample rate as high as $1/O(\log n)$, it can produce (via [3]), for example, a polynomial-time quantum algorithm for SVP with an approximation factor $\sqrt{n}$ polylog(n), or for LWE instances with $\alpha = \sqrt{n}$ polylog(n) [11] [9].

2 The Algorithm

2.1 Algorithm Overview

Regev's polynomial-time DSP algorithm [12] that uses a subset sum oracle can be summarized as follows: first, generate slightly more than n DSP samples $\sum_{b_i \in \{0,1\}} |b_i, x + b_i d\rangle$ (all arithmetic operations being $\mod N$, where $N \approx 2^n$), then Fourier-transform the $x + b_i d$ portion of each and measure it, producing a superposition of 0 and 1 and a measured value y_i, such that the phases of 0 and 1 differ by a factor $\omega^{y_i d}$ (where ω is an Nth root of unity). Next, compute the subset sum $\sum b_i y_i$ in superposition, and measure all but the most significant bit h of it, calling the measured bits z'. The phases of all of the states that survive this measurement will be $\omega^{\sum b_i y_i d} = \omega^{(z' + hN/2)d}$, where $\omega^{z' d}$ is constant. We can then use the subset sum oracle to compute (in superposition) the solutions to the two subset sum problems $\sum b_i y_i = z' + hN/2$ given z' for $h = 0$ and $h = 1$. With non-negligible probability, there will be exactly one solution for each value of h, which can be used to "erase" the b_i values in each case, leaving only h distinguishing the two solution states. Their phases will differ by $\omega^{Nd/2}$, which will be 1 if the last bit d_n of d is 0 and -1 if $d_n = 1$. Hence

Hadamard-transforming h and measuring it will produce d_n with probability 1. The remaining bits of d can be obtained by repeating this algorithm recursively, using knowledge of d_n to erase the last bit from every sample and obtaining d_{n-1} as above, and so on.

We will follow this outline, but use a different method to erase the b_i values. First, as before, we will generate DSP samples $\sum_{b_i \in \{0,1\}} |b_i, x + b_i d\rangle$ (some of them faulty—i.e., a random $(n+1)$-bit value, not in superposition), Fourier-transform the $x + b_i d$ portions y_i, and compute the subset sum z in superposition, measuring the least significant $n-1$ bits z' of the sum, while leaving the most significant bit h unmeasured. However, we will do this with kn^{c+1} samples (for constants k and c), forming a superposition of states $\phi = b_1 ... b_{kn^{c+1}}$. We then process the bits in groups of $c \log n$, as follows: first, we compute in superposition the $\log n$ most significant bits s of the sum $r_j = \sum b_i y_i$ for group g_j, then we Hadamard-transform the b_i values in g_j and measure them along with the corresponding y_i values. If all the b_i measurements result in 0, then we include this group in set A—otherwise, we will measure all of the bits of s for the group, and then include it in set B.

Our goal will be to run Regev's algorithm on A—which has no extraneous phases introduced by the erasure of its bits b_i—instead of on the entire set of samples, using the s values for A to construct a new h^* that replaces the original h, and transferring the phase that encodes d_n from h to h^*. The lower-order $n-1$ bits of the replacement subset sum $z^* = \sum_A b_i y_i$ over just A won't ever be explicitly calculated—it'll range over all possible values, in fact—but it'll be used implicitly to argue that the samples in B have an identical effect regardless of the value of h^*, and hence can be ignored.

We can successfully collect an expected $a = n/\log n$ groups in set A, assuming $k > c$ and thus enough samples that an overwhelmingly large fraction of the possible measured strings contain at least that many groups qualifying for set A. We then compute the sum s^* of the s values, measure all but the last s value s_a, and erase s_a using s^* and the other measured s values. Next, we Hadamard-transform and measure all the bits of s^* except the most significant bit h^*. Assuming we get the desired all-zeroes result (with expected probability $2/n$ over choices of the y_i's), we're left with the single bits h and h^*, where h^* divides all the states based on whether z^* has highest-order bit 0 or 1. (Because we've used the most significant $\log n$ bits of the sums for each group, and there are only $n/\log n$ s values summed to obtain s^*, the computed value of h^* will be the correct highest-order bit of z^*, as long as the most significant $\log \log n$ bits of s^* contained at least one 0—a condition we can check for.)

Finally, we compute and measure $h' = h \oplus h^*$. Recall that h starts off encoding d_n in the Hadamard basis—that is, that the $h = 0$ and $h = 1$ states have the same or opposite phases depending on whether $d_n = 0$ or $d_n = 1$. If for each value of h' the effect of Hadamard-transforming the b_i's in set B on the phases of $h^* = 0$ and $h^* = 1$ is the same, then it's easy to verify that the computation of h' effectively copies h's encoding of d_n to h^*. (Whether the bits of h and h^* are exactly the same or exactly opposite, the relationship between the final phases of their two states will be the same for both qubits.) We can thus use h' and h^* to erase h, leaving a single unmeasured bit h^*, and measure h^* in the Hadamard basis, in place of h, to obtain d_n. All that remains is to show that the effect of set B on the amplitudes and phases of the states $h^* = 0$ and $h^* = 1$ is approximately equal for both.

Now, these $h^* = 0$ and $h^* = 1$ states are each composed of the sum of all the original states ϕ (each with phase 1 or -1) consistent with the measured values and that value of h^*. We can consider each ϕ to be of the form (ϕ_A, ϕ_B), where ϕ_A consists of the portion of the original state associated with groups of samples in A, and similarly for ϕ_B. We can then partition these state-portions ϕ_A and ϕ_B into sets based on their corresponding value of z^*. We make the following observations:

(1) For any particular value of $\bar{z^*}$, the least significant $n-1$ bits of z^*, the same set of ϕ_B state-portions are consistent with $\bar{z^*}$ and h^* irrespective of the value of h^*. For example, if h' was

measured to be 0 and ϕ_B is consistent with the value of $\bar{z^*}$ and $h^* = 0$, then it will also be consistent with $\bar{z^*}$ and $h^* = 1$, because adding 2^{n-1} to both z^* and z' won't change their difference $\pmod N$. The phases of the sums of the states ϕ consistent with $\bar{z^*}$ for both values of h^* will thus be the same, once h' is computed and measured, and the contributions of the ϕ_B state portions to each amplitude will also be identical for both values of h^*.

(2) Let z_h^* be the most significant $\log n$ bits of z^*, and z_l^* be the remaining bits. Then for a sufficiently large constant c and any z_h^*, the state-portions ϕ_A are extremely close to uniformly distributed across values of z_l^*. This is a result of the pairwise independence of the subset sums z_l^* [12]. (There are approximately $2^{(cn(1-1/O(\log n))-n-c\log n}$ ϕ_A values distributed over the $2^{n-\log n}$ possible z_l^* values—assuming a $1/O(\log n)$ faulty sample rate—so the expected $2^{(cn(1-1/O(\log n))-2n-O(\log n)}$ states per value will vary by a standard deviation which is the square root of that number, and which can be made exponentially small compared to the total—even when totaled over all the possible values of z_l^*—by choosing a large enough c.)

(3) Moreover, for any particular pair of values of z_h^* that differ only in h^*, the number of ϕ_A state-portions consistent with each is approximately equal—differing by an expected fraction about $1/n^{(c-1)/2}$ of the total. This is again a result of the pairwise independence of subset sums, this time in the group a that produces s_a. (We consider here only the randomness in z_h^* associated with the generation of s_a—the additional randomness stemming from "overflow" bits generated by summing the lower $n - \log n$ bits of the subset sum only increases the entropy, and hence the uniformity, of z_h^*. We also assume that g_a's $c \log n$ samples are all non-faulty, which is true with probability approximately $e^{-c/c'}$, where the faulty sample rate is $1/c' \log n$.)

It follows that for any pair of values of z_h^* differing only in h^*, the expected difference in the sizes of their amplitudes will be on average about $1/n^{(c-1)/2}$—since the amplitudes for pairs of corresponding z_l^* values will all be almost exactly the same, by observations 1 and 2, and with the same phase, by observation 1—and hence the total expected difference in the magnitudes of the amplitudes of $h^* = 0$ and $h^* = 1$ will average about $1/n^{(c-3)/2}$. (We're assuming here that the amplitudes associated with each z^* value are "well-behaved" enough—that is, that they aren't so huge and mutually canceling—that the tiny differences among amplitudes associated with the ϕ_A state-portions for different z^* values aren't unduly magnified. Fortunately, the pairwise independence of the phases of states over possible measured values of the Hadamard-transformed bits b_i ensures that these per-z^* amplitudes are within reasonable bounds with high probability.) For large enough c, then, this total difference in amplitude will be small enough that we can (by repeating the algorithm multiple times) recover d_n from h^* as we did from h in the original algorithm, and recurse as before to recover all of d.

2.2 Algorithm Details

(Note: The proofs in this section are presented as sketches—more formal proofs can be found in [6].)

Theorem. There exist a polynomial-time (in n) quantum algorithm and constant c' such that, given DSP samples of the form $\sum_{b_i \in \{0,1\}} |b_i, x + b_i d \pmod N\rangle$ (where N is exponential in n and x is a random value $\pmod N$) with probability at least $1 - 1/c' \log n$, and a random $(n+1)$-bit value otherwise, computes the last bit d_n of d.

Proof. For simplicity, let $N = 2^n$, and assume all arithmetic operations and relations involving states, amplitudes and phases are $\pmod N$. We will describe each step of the algorithm in sequence, along with the expression representing the superposition created at that step.

In Step 1, we collect a sequence of $Q = kn^{c+1}$ $(n+1)$-bit DSP samples $R_i = 2^{-1/2}\sum_{b_i\in\{0,1\}}|b_i, x_i + b_i d\rangle$ (or, in the case of faulty samples, $R_i = |b_i, x_i\rangle$). For each sample, we Fourier-transform the $x_i + b_i d$ portion, producing the values $y_1...y_Q$, and move the y_i values to the end, creating the concatenated string Y. The resulting superposition is as follows:

$$2^{-(n+1)Q/2}\sum_{b_1...b_Q,Y}\omega^{\sum_i y_i(x_i+b_i d)}|b_1, ..., b_Q, Y\rangle$$

In step 2, we compute the sum $z = \sum_{i=1}^{Q} b_i y_i$ in superposition, and measure all but the highest-order bit of the result, obtaining the $(n-1)$-bit value z'. We'll define Z'_Y as the set of vectors $\phi = b_1...b_Q$ instantiated by the samples and consistent with Y and the measured value of z' (for one of $h = 0$ or $h = 1$), and h as the highest-order bit of z. The resulting superposition is thus as follows:

$$(\sum_Y |Z'_Y|)^{-1/2}\sum_{Y,\phi\in Z'_Y}(-1)^{hd_n}\omega^{\sum_i x_i y_i + z' d}|\phi, Y, h, z'\rangle$$

Henceforth we'll omit the measured value z' and the constant value $\omega^{z' d}$ from the superposition.

In step 3, we divide the bits b_i into groups g_j of size $c\log n$, and compute the value s_j as the most significant $\log n$ bits of the sum $r_j = \sum_{g_j} b_i y_i$. The resulting superposition is as follows:

$$(\sum_Y |Z'_Y|)^{-1/2}\sum_{Y,\phi\in Z'_Y}(-1)^{hd_n}\omega^{\sum_i x_i y_i}|\phi, Y, h, s_1...s_{Q/c\log n}\rangle$$

In step 4, we Hadamard-transform the bits b_i of each group g_j, producing the following superposition over all strings $D = b'_1...b'_Q$:

$$2^{-Q/2}(\sum_Y |Z'_Y|)^{-1/2}\sum_D\sum_{Y,\phi\in Z'_Y}(-1)^{hd_n+\sum b_j b'_j}\omega^{\sum_i x_i y_i}|D, Y, h, s_1...s_{Q/c\log n}\rangle$$

We then measure Y (renaming Z'_Y for the measured Y as Z') and D, and move the measured bits of the first $a = n/\log n$ groups for which the b'_i are all 0 to the front (i.e., henceforth labeling them $b_1...b_{cn}$). We call these latter groups (and their associated s_j values) set A, and include the remaining $\beta = (Q/c\log n) - (n/\log n)$ groups in set B. We'll also henceforth omit the now-constant phase coefficient $\omega^{x_i y_i}$.

Lemma 1. If $k > c$, then with constant probability, the measured value of $D = b'_1...b'_Q$ will have at least $n/\log n$ groups that consist of all zeroes.

Proof. (Sketch) Let D_0 be the set of D values with more zeroes than ones, let D_1 be the set of D values with more ones than zeroes, and let $D_{0=1}$ be the remainder (with an equal number of ones and zeroes). Also, let Z_{even} be the set of $\phi \in Z'$ with an even number of ones, and Z_{odd} be $Z' - Z_{even}$. The portion of the superposition corresponding to D_1 after the measurement of Y is

$$2^{-Q/2}|Z'|^{-1/2}\sum_{D_1}\sum_{\phi\in Z'}(-1)^{hd_n+\phi\cdot D}|D...\rangle$$

$$= 2^{-Q/2}|Z'|^{-1/2}(\sum_{D_1}\sum_{\phi\in Z_{even}}(-1)^{hd_n+\phi\cdot D}|D...\rangle + \sum_{D_1}\sum_{\phi\in Z_{odd}}(-1)^{hd_n+\phi\cdot D}|D...\rangle)$$

$$= 2^{-Q/2}|Z'|^{-1/2}(\sum_{D_0}\sum_{\phi\in Z_{even}}(-1)^{hd_n+\phi\cdot D}|D...\rangle - \sum_{D_0}\sum_{\phi\in Z_{odd}}(-1)^{hd_n+\phi\cdot D}|D...\rangle)$$

since the phase $(-1)^{\phi\cdot D}$ for any D and its counterpart with all the bits reversed will be identical for Z_{even}, and opposite for Z_{odd}. Given that the sums are all as likely (over choices of Y) to be negative as positive, and as least as likely to have opposite phases as identical ones between Z_{even}

and Z_{odd}, it follows that the expected total probability (over choices of Y) of D_0 is at least as large as that of D_1. Hence with at least constant probability, D has at least as many zeroes as ones.

Now, consider the set DY of pairs (D, Y) such that $D \in D_0 \cup D_{0=1}$ and Y consists of values y_i such that the lower-order $n - 2\log n$ bits of the y_i values are all distinct. (The latter condition holds for the overwhelming majority of possible Y, and thus DY still covers a constant fraction of the total probability space.) Let DY_{bad} be the set of elements of DY that do not produce the required $n/\log n$ all-zero groups, and DY_{good} be $DY - DY_{bad}$. We can map every element of DY_{bad} to a distinct element of DY_{good} as follows: we first swap locations of zero bits and one bits in D so that the new D is in DY_{good}, then we swap the lower $n - 2\log n$ bits of the corresponding y_i values. The values of ϕ that are in Z'_Y will also have their bits rearranged the same way as those of D, and the phase of each such ϕ will be unchanged after the rearrangement, given the rearrangement of the bits of D. The total probability of these mapped elements of DY_{good} will therefore be as large as the probability of DY_{bad}, and DY_{good} will thus have an expected probability at least as large as DY_{bad}. It follows that with at least constant probability, the measured value (D, Y) will be in DY_{good}. □

The resulting superposition (assuming the required number of all-zero groups) is as follows:

$$\nu_1 \sum_{\phi \in Z'} (-1)^{hd_n + \sum_B b_j b'_j} |0^{cn} b'_{cn+1}, ..., b'_Q, Y, h, s_1 ... s_{Q/c \log n}\rangle$$

where ν_1 is the normalization coefficient determined by the measurement of Y and D. Henceforth we'll omit the measured values Y and (rearranged) D from the superposition.

We can rewrite this superposition in terms of the sum $z^* = \sum_A r_j = \sum_{i=1}^{cn} b_i y_i$—that is, defining the sets Z'_{z^*} of states $\phi \in Z'$ for which $\sum_{i=1}^{cn} b_i y_i = z^*$. We thus create the following superposition:

$$\nu_1 \sum_{z^*} \sum_{\phi \in Z'_{z^*}} (-1)^{hd_n + \sum_{j=cn+1}^{Q} b_j b'_j} |h, s_1 ... s_{Q/c \log n}\rangle$$

Now, consider the set A_{z^*} of strings $\phi_A = b_1 ... b_{cn}$, $\phi_A \in Z'$ such that $\sum_{i=1}^{cn} b_i y_i = z^*$, and the set $B_{z^*,h}$ of strings $\phi_B = b_{cn+1} ... b_Q$, $\phi_B \in Z'$ such that $\sum_{i=cn+1}^{Q} b_i y_i + z^* = z' + 2^{n-1}h$ for $h = 0$ or $h = 1$. Any $\phi \in Z'_{z^*}$ must consist of a concatenation of an element of A_{z^*} and an element of $B_{z^*,h}$ (for one of $h = 0$ or $h = 1$), and moreover any such concatenation must be in Z'_{z^*}. We can hence rewrite the above superposition as follows:

$$\nu_1 \sum_{z^*,h} \sum_{\phi_A \in A_{z^*}} \sum_{\phi_B \in B_{z^*,h}} (-1)^{hd_n + \sum_{j=cn+1}^{Q} b_j b'_j} |h, s_1 ... s_{Q/c \log n}\rangle$$

In step 5, we measure the values s_i for groups in B, leaving values $S = (\sigma'_1 ... \sigma'_\beta)$. We define the sets $B_{z^*,S,h}$ as the sets of states in $B_{z^*,h}$ consistent with the measured value S. The resulting superposition is as follows:

$$\nu_2 \sum_{z^*,h} \sum_{\phi_A \in A_{z^*}} \sum_{\phi_B \in B_{z^*,S,h}} (-1)^{hd_n + \sum_{j=cn+1}^{Q} b_j b'_j} |h, s_1 ... s_a, \sigma'_1 ... \sigma'_\beta\rangle$$

where ν_2 is the revised normalization coefficient determined by the measurements. For brevity, we define

$$C_{z^*,S,h} = \sum_{\phi_B \in B_{z^*,S,h}} (-1)^{\sum_{j=cn+1}^{Q} b_j b'_j}$$

and rewrite the above superposition as

$$v_2 \sum_{z^*,h} \sum_{\phi_A \in A_{z^*}} (-1)^{hd_n} C_{z^*,S,h} |h, s_1 ... s_a, \sigma'_1 ... \sigma'_\beta\rangle$$

Henceforth we'll omit the measured values $\sigma'_1 ... \sigma'_\beta$.

In step 6, we compute the sum $s^* = \sum_j s_j \pmod n$, using $s_1 ... s_{a-1}$ and s^* to erase s_a, and compute the bit l_{s^*} which is 1 if the second-most significant through $\log\log n$-most significant bits of s^* are all 1, and 0 otherwise. We then Hadamard-transform $\bar{s^*}$ (the low-order $\log n - 1$ bits of s^*), and measure $W = (s_1 ... s_{a-1}, l_{s^*})$, producing the measured value W'. We proceed only if the measured value of l_{s^*} is 0 (an event which occurs with at least constant probability, by the same reasoning as Lemma 1). We also measure the Hadamard-transformed bits of $\bar{s^*}$, and proceed only if the result is all zeroes—an event which occurs with expected probability $2/n$, over choices of Y.

We define the sets $A'_{z^*,W'}$ as the sets of states in A_{z^*} consistent with the measured value W'. We also label the most significant bit of s^* as h^*, giving us the following superposition:

$$v_3 \sum_{z^*,h} \sum_{\phi_A \in A_{z^*,W'}} (-1)^{hd_n} C_{z^*,S,h} |h, h^*, W', 0^{\log n - 1}\rangle$$

where v_3 is the revised normalization coefficient determined by the measurements. Note that h^* as calculated is the most significant bit of the sum of the most significant $\log n$ bits of the $n/\log n$ values r_j whose sum is z^*, and also that $l_{s^*} = 0$ (meaning that the most significant $\log\log n$ bits of s^*, apart from h^*, aren't all 1). We can therefore be sure that the omitted less significant bits of the subset sums r_j don't affect the computation of h^*, and hence that h^* correctly represents the most significant bit of z^*. Henceforth we'll omit the measured values W' and $0^{\log n - 1}$.

Finally, in step 7, we compute and measure $h' = h \oplus h^*$, and use it to erase h. We also replace $C_{z^*,S,h}$ with $C_{z^*,S,h'}$, which is defined equivalently to $C_{z^*,S,h}$ but with $B_{z*,S,h}$ replaced by $B'_{z*,S,h'}$, the same set of states restricted to those elements consistent with the measured value of h'. We thus produce the following superposition:

$$v_4 \sum_{z^*} (-1)^{(h^* \oplus h')d_n} \sum_{\phi_A \in A_{z^*,W'}} C_{z^*,S,h'} |h^*, h'\rangle$$

where v_4 is the revised normalization coefficient determined by the measurement of h'. We will now show that $\sum_{\phi_A \in A_{z^*,W'}} C_{z^*,S,h'}$ will be approximately equal for either value of h^*, given two values of z^* whose least significant $n-1$ bits (which we'll call $\bar{z^*}$) are equal.

Lemma 2. *$C_{z^*,S,h'}$ is equal for any two values z^*_0 and z^*_1 of z^* that differ only in the most significant bit h^*, and either measured value of h'.*

Proof. For any element ϕ_B of $B'_{z^*_0,S,h'}$, $z^*_0 + \sum_B b_i y_i = z' + 2^{n-1}h$. We can add or subtract 2^{n-1} as necessary from both z^*_0 and $z' + 2^{n-1}h$ to obtain the equality $z^*_1 + \sum_B b_i y_i = z' + 2^{n-1}h$ for the other value of h^* (and hence of h, since we've fixed $h' = h^* \oplus h$ by measuring it). Thus $B'_{z^*_0,S,h'} = B'_{z^*_1,S,h'}$, and the lemma's claim follows. □

We can therefore rewrite the above superposition in terms of $C_{\bar{z^*}} = C_{z^*,S,h'}$ for either z^* whose least significant $n-1$ bits are $\bar{z^*}$ (and the measured value of h'), as follows:

$$\psi = v_4 \sum_{z^*} (-1)^{(h^* \oplus h')d_n} \sum_{\phi_A \in A_{z^*,W'}} C_{\bar{z^*}} |h^*, h'\rangle$$

We next show that the portion of the amplitude of either value of h^* that is associated with a particular z^* is of bounded size. This will allow us later to convert the bounded relative deviations

from uniformity of these amplitude portions into a bounded absolute total deviation across all values of z^*.

Definition 1. For a given measured z', and a resulting superposition ψ of values of h^* at the end of step 7, let M be the set of values (Y, D, W', S, h') measured (apart from z') during the algorithm, let T_M^+ (resp., T_M^-) be the set of states ϕ consistent with M and h^* and with phase 1 (resp., -1) resulting from measurement of M (ignoring the constant phase $\omega^{yz'd}$). Let $T_M = T_M^+ \cup T_M^-$, and let t_M (resp., t_M^+, t_M^-) be $|T_M|$ (resp., $|T_M^+|$, $|T_M^-|$).

Lemma 3. The probability (over choices of M) that for some z^* the amplitude $\alpha_{z^*} = \nu_4 \sum_{\phi_A \in A_{z^*,W'}} C_{\bar{z^*}}$ has magnitude greater than $\Omega(2^n)$ is less than $O(2^{-n})$.

Proof. (Sketch) Let $T_{M'}$ (resp., $T_{M'}^+$, $T_{M'}^-$) be defined identically to T_M (resp., T_M^+, T_M^-), but with $\bar{h'}$ replacing h', and let $T = T_M \cup T_{M'}$, $T^+ = T_M^+ \cup T_{M'}^+$, and $T^- = T_M^- \cup T_{M'}^-$. Note that T_M and $T_{M'}$ are disjoint and of equal expected size $t/2$ (where $t = |T|$) over choices of Y, since for any Y that maps a given ϕ to T_M, there's a corresponding Y' that maps it to $T_{M'}$, where the difference is only the addition of 2^{n-1} to two y_i values, one on either side of the boundary between ϕ_A and ϕ_B—say, in g_a on the ϕ_A side, and among the unused samples on the ϕ_B side, to maintain consistency with measured s_i values. Note that this partition depends only on Y, and not on D. (All values of D are equally compatible with any partition of ϕ values into T_M and $T_{M'}$—D only determines the ϕ values' phases.)

Now, consider a set M_D of M values with fixed (Y, W', S, h') and varying D. T_M and $T_{M'}$ will be the same for all values of D in M_D, since any D is compatible with any ϕ value, and only affects the phase of ϕ. Moreover, any ϕ (resp., any pair ϕ_1 and ϕ_2), regardless of whether in T_M or $T_{M'}$, is equally often in T^+ or T^- (resp., both in T^+, both in T^-, or one of them in each), over choices of D. (This holds as long as there's a single non-zero bit in ϕ, since that bit leaves the phase of ϕ unchanged if paired with a zero in D—that is, for half the possible values of D—and flips ϕ's phase if paired with a 1 in D. The exception, the all-zeroes state, only appears in the superposition if the measured value of z was 0, which occurs with exponentially small probability.) The phases of the elements of T_M can thus be considered to be selected pairwise-independently over choices of D (assuming uniform D—that is, not considering the actual probability that each D is measured). The sizes of T_M^+ and T_M^-—that is, t_M^+ and t_M^-—will therefore have mean and variance $\sigma_M^2 = \Theta(t_M)$.

Moreover, the subsets T_{M,z^*}^+ and T_{M,z^*}^- of T_{M,z^*} (where T_{M,z^*} is defined for each D as the set of $\phi \in T_M$ compatible with z^*) for a given z^* can be partitioned according to the value of their ϕ_B state portions (which define their phase). And since the distinct ϕ values in each such partition must differ within their ϕ_A state portions, they are therefore assigned z^* values pairwise independently over choices of Y. Hence the pairwise independence of phases of ϕ states over choices of D (and Y), and the resulting mean and variance of t_M^+ and t_M^-, carry over to the sizes t_{M,z^*}^+ and t_{M,z^*}^- of T_{M,z^*}^+ and T_{M,z^*}^-, respectively: t_{M,z^*}^+ and t_{M,z^*}^- will have mean and variance $O(t_M/2^n)$.

We next consider the expected value $E(\alpha_{z^*}^2)$ for some z^* over (the actual distribution on) D in a set M_D and a random choice of Y. It can be expressed as $\sum_D Pr[D]\alpha_{z^*,D}^2$, where $\alpha_{z^*,D}$ is the portion of D's amplitude post-normalization that corresponds to states ϕ compatible with z^*—i.e.,

$$\alpha_{z^*,D} = (t_M \cdot |M_D| \cdot Pr[D])^{-1/2} |t_{M,z^*}^+ - t_{M,z^*}^-|$$

Hence

$$E(\alpha_{z^*}^2) = \sum_D (t_{M,z^*}^+ - t_{M,z^*}^-)^2/(t_M \cdot |M_D|) = E_D((t_{M,z^*}^+ - t_{M,z^*}^-)^2/t_M)$$

where E_D is the expectation over the uniform distribution on D. Note that $E_D((t_{M,z^*}^+ - t_{M,z^*}^-)^2)$ is the variance of $t_{M,z^*}^+ - t_{M,z^*}^-$, and is therefore $O(t_M/2^n)$, as computed above for t_{M,z^*}^+ and t_{M,z^*}^-. It follows that $E(\alpha_{z^*}^2)$ is $O(2^{-n})$, and hence the probability that $\alpha_{z^*}^2 > \Omega(2^n)$ (meaning $\alpha_{z^*} > \Omega(2^{n/2})$)

is at most $O(2^{-2n})$. The probability that for at least one value of z^* $\alpha_{z^*} > \Omega(2^{n/2})$ is thus at most $O(2^{-n})$. □

Corollary. For any $\log n$-bit value z_h^* let Z_h^* be the set of values of z^* whose most significant $\log n$ bits are z_h^*, and let $\alpha_{z_h^*}$ be the amplitude of z_h^* in the superposition, i.e., $\sum_{Z_h^*} \alpha_{z^*}$, where α_{z^*} is defined as in Lemma 3. Then the probability (over choices of M) that for some z_h^* consistent with h^* the amplitude $\alpha_{z_h^*}$ has magnitude greater than $n^{1/2}$ is less than $O(1/n)$.

Proof. (Sketch) The proof follows the structure of Lemma 3, but with z_h^* replacing z^* and n substituting for 2^n (and hence $\sqrt{n}$ substituting for $2^{n/2}$). □

Next, we will show that for two z_h^* values that differ only in h^*, their implicit amplitudes in ψ following step 7 are approximately equal. We do so by demonstrating that the ϕ_A state portions are so close to uniformly distributed over z^* values that, given the bound established by Lemma 3, the amplitudes of any two z^* values differing only in h^* are close to identical—so close, in fact, that their sums across all z^* values sharing the same z_h^* (apart from h^*) are also approximately identical.

Lemma 4. For any z^*, let z_h^* be the most significant $\log n$ bits of z^*, and z_l^* be the remaining bits. Then for any two values $z_{h,0}^*$ and $z_{h,1}^*$ of z_h^* that differ only in the most significant bit h^*, and for $c \geq 12$, the amplitudes $\alpha_{z_{h,0}^*}$ and $\alpha_{z_{h,1}^*}$ (defined as in the corollary to Lemma 3) differ by an expected multiplicative factor $1 \pm O(n^{-(((c-1)/2)-1)})$ with probability at least $1 - 1/n$.

Proof. (Sketch) For any ϕ_A, the value of z_h^* is the sum of the "overflow" bits produced by summing the relevant z_l^* values, the measured s_j values, and s_a. We will focus on the latter, dividing the state-portions ϕ_A into sets A_{g_a}, each with a different value of the string q_{g_a} of values of the bits in g_a, and assuming that z_h^* is fully determined by s_a. (Considering the additional variation from the "overflow" bits in determining z_h^* would only increase the uniformity of z_h^*, which we want to maximize in any event.) Note that the sets A_{g_a} are of equal size, since a particular value of q_{g_a} doesn't restrict the values of the rest of the bits in ϕ_A, as long as z^* isn't specified.

Since the values of s_a for ϕ_A in different A_{g_a} are pairwise independent over all possible y_i values, we can model the distribution of z_h^* values—based solely on the variation in s_a—as a "balls in bins" problem, with n^c balls and n bins. The mean for each bin is thus n^{c-1} balls, with a variance n^{c-1} and standard deviation $n^{(c-1)/2}$ for each bin. (We are assuming here that g_a contains no faulty samples—an event which occurs with probability greater than $(1 - 1/c' \log n)^{c \log n} = e^{-c/c'}$.) By Chebyshev's inequality, then, the probability that the number of balls in a particular bin differs from n^{c-1} by $\kappa n^{(c-1)/2}$ is at most $1/\kappa^2$, and the probability that any bin deviates from the mean by $\kappa' n^{(c-1)/2}$ is thus at most n/κ'^2. For example, for $\kappa = n$, the bins deviate from the mean by at most $n^{((c-1)/2)+1}$ except with probability at most $1/n$.

The values of z_l^* for different ϕ_A are also pairwise independent, so we can again model the distribution of z_l^* values for each A_{g_a} as a "balls in bins" problem, with an expected $\tau = 2^{cn(1-1/(c' \log n))-n-c \log n}$ balls (since the $n - \log n$ measured bits of W', plus the $\log n$ bits of z_h^*, reduce the cn bits of A by an expected n effective bits, and g_a has $c \log n$ bits) distributed among $2^{n-\log n}$ bins. (To simplify the calculations we'll consider the number of bins as 2^n.) The mean number of balls per bin is thus $\mu = 2^{cn(1-1/(c' \log n))-2n-c \log n}$, with an equivalent variance and thus a standard deviation $\sigma = 2^{(cn/2)(1-1/(c' \log n))-n-c \log n/2}$.

By Chebyshev's inequality, then, the number of balls will deviate from μ by $\kappa'\sigma$ for a given bin with probability at most $1/\kappa'^2$. Setting $\kappa' = 2^n$ and $c \geq 12$, we get that with probability at least $1 - 2^{-2n}$, a bin deviates from μ by at most $2^{(12n/2)(1-1/(c' \log n))+n-12 \log n/2}$ balls—or, in amplitude terms, a z_l^* value deviates by that much from the expected number of ϕ_A state portions. We know from Lemma 3 that the amplitude of a given (z_h^*, z_l^*) pair has (except with exponentially small probability) magnitude at most $2^{n/2}$. It follows that with probability at least $1 - 2^{-n}$, the total

deviation δ from μ over all $2^n - n$ possible values of z_l^* for a given z_h^* is at most

$$\begin{aligned}&((2^{(12n/2)(1-1/(c'\log n))+n-12\log n/2})/\mu)(2^{n/2})(2^n)\\ &= (2^{(12n/2)(1-1/(c'\log n))+5n/2-12\log n/2})/\mu\\ &< 2^{-n}\end{aligned}$$

even if all the relative deviations are maximal and all in the same direction.

We further observe from the superposition ψ yielded by step 7 that the amplitude associated with a given set A_{g_a} is given by

$$\nu_4 \sum_{z_l^*} (-1)^{(h^* \oplus h')d_n} \sum_{\phi_A \in A_{(z_h^*, z_l^*)}, W' \cap A_{g_a}} C_{(z_h^{*-}, z_l^*)}$$

Because of the nearly-uniform distribution of the ϕ_A state-portions over values of z_l^*, though, we can treat this amplitude as a simple sum of the relevant $C_{\vec{z^*}}$ values, multiplied by the expected number μ of compatible state-portions ϕ_A, ignoring the resulting exponentially small error δ. That is, we can approximate the amplitude of A_{g_a} as

$$\nu_4 (-1)^{(h^* \oplus h')d_n} |A_{g_a}| \sum_{z_l^*} C_{(z_h^{*-}, z_l^*)}$$

where the sets A_{g_a} are of equal size. Moreover, for two q_{g_a} values whose corresponding z_h^* values differ at most in h^* the $C_{\vec{z^*}}$ corresponding to each z_l^* is the same (by Lemma 2)—as is, therefore, the sum over all z_l^* of the $C_{\vec{z^*}}$. Hence if we define $J = \nu_4 |A_{g_a}| \sum_{z_l^*} C_{(z_h^{*-}, z_l^*)}$ for a given z_h^*, then the amplitude of z_h^* is just J multiplied by the number of sets A_{g_a} corresponding to the given z_h^*. The relative multiplicative difference between two values of z_h^* differing only in h^* is thus bounded (with probability at least $1 - 1/n$) by $1 \pm O(n^{((c-1)/2)+1-(c-1)}) = 1 \pm O(n^{-(((c-1)/2)-1)})$, as required. □

Using Lemma 4 along with the corollary to Lemma 3, it follows that the amplitudes of the values of z_h^* in the superposition ψ following step 7 are all at most $\sqrt{n}$ except with probability at most $O(1/n)$, and hence for $c \geq 12$ the sum of the differences in amplitudes across the $n/2$ pairs z_h^* that differ only in h^* is at most $O(n^{-(((c-1)/2)-5/2)}) \leq 1/n$ except with probability $1/n$. This is small enough compared to the approximately $1/\sqrt{2}$ amplitude of at least one of the two possible values of h^* (and hence both, since their difference is so small) that Hadamard-transforming and measuring h^*, which has inherited its encoding of d_n from h by the computation of h', will yield d_n with probability significantly greater than $1/2$. Thus polynomially many iterations of the algorithm will allow us to extract d_n with near-certainty. □

Corollary. There exists a polynomial-time quantum algorithm that computes $O(\sqrt{n}\,\mathrm{polylog}(n))$-factor approximations of SVP solutions, and solutions to LWE instances with $\alpha = O(\sqrt{n}\,\mathrm{polylog}(n))$.[11][9]

3 A Note on [7]

Gupte, Ragavan and Zhandry [7] recently presented a proof that the density matrix representing the state of the above algorithm following step 2—assuming averaging over untouched portions of the y_i values—is exponentially close to that which would be produced if all the qubits b_i were traced out (i.e., measured and then discarded). They argue that this result implies that the above algorithm can't possibly produce a correct value of d_n with non-negligible probability, since the information about d_n is encoded in the relationship between the phases of each value of h, and thus destroyed when all the b_i's are traced out, collapsing h to a single value.

As it turns out, Lemmas 1, 2 and 3 are all completely compatible with the result in [7]—Lemmas 2 and 3, in particular, focus on the ϕ_B portion of the state, which could in fact be traced out, reducing it to a single value, without affecting the algorithm at all. Rather, it's Lemma 4—–the claim that

the ϕ_A state portions are roughly evenly distributed among z^* values—that fails once the bits in ϕ_A are traced out.

Now, this would certainly cause the algorithm to fail—set A is exactly the set that needs to be represented for both values of h and h^*, and hence not be traced out, for the algorithm to work. However, the fact that A is post-selected such that the results of the Hadamard transforms on its bits are all zero allows the states of h and h^* computed in the algorithm—perhaps surprisingly—to diverge non-negligibly from the case where the values of the b_i's in A are traced out, despite the theorem proven in [7].

Consider, for example, the case where *all* the b_i's are measured as zero following their Hadamard transforms. (Or alternatively, consider a variation of the algorithm in which a random set A of $n/(\log n)$ groups of b_i's is pre-selected, then the rest of the b_i's are measured, hoping that the Hadamard transforms on set A all result in zeroes.) In that case, the algorithm can easily be seen to work correctly (as a mildly modified version of Regev's original algorithm, with the b_i's just erased a different way), even though its pre-Hadamard-transform density matrix, per [7], is only negligibly different from the one for the case where the b_i's were all traced out, preventing the algorithm from working correctly. Of course, the Hadamard transform on the b_i's (or in the variation, the b_i's in A) only results in all zeroes with exponentially small probability. But note that the probability expressed in the density matrix in [7] is over choices of the y_i's, and an all-zeroes outcome of the Hadamard transforms makes the *average* output across all y_i's completely different from the all-b_i's-traced-out case. In effect, the Hadamard transform concentrates the expected difference between the intact-b_i's and traced-out-b_i's cases in the all-zeroes outcome, making it non-negligible when conditioned on that outcome occurring.

(In density matrix terms, the effect of the negligibly-different density matrices on the states of the computed values h^* and h normally carries through the Hadamard transform because the states whose amplitudes are summed have varying phases, and hence interfere negatively, keeping the difference small. But this effect breaks down when the result of the Hadamard transform is all zeroes, and therefore imparts no phases to the input states, causing the two density matrices, when h^* and h are included, to diverge significantly.)

This exceptional behavior of the all-zeroes Hadamard transform result is important because the crux of the argument in [7] is the implicit claim that the negligible difference between the actual and traced-out density matrices over choices of the untouched portions of the y_i's is persistently uniform and completely independent of the evolution of the algorithm's state. But this claim is in fact incorrect, as we've shown, since the difference can become non-negligible in the case of an all-zeroes Hadamard transform result on a given set A of b_i's. Moreover, this case is recognizeable to the algorithm, and can even be amplified and exploited by it. And that's precisely what the algorithm does by post-selecting A to have an all-zeroes Hadamard transform result: the negligible probability of this event occurring—and the density matrix thus diverging from the traced-out case, yielding a correct algorithm output—for any particular choice of A instead becomes a near-certainty when A is post-selected based on the outcome of the Hadamard transforms on the b_i's.

More precisely, consider another variation of the algorithm, identical to the previous one in which A is randomly pre-selected, except that we omit the measurement of the b_i's in set B (still hoping that the Hadamard transforms on the b_i's in A all result in zeroes). The proof presented in Section 2 can easily be adapted to show that this variation of the algorithm works correctly in the (exponentially unlikely) case that the Hadamard transforms on the b_i's in the pre-selected A do in fact all produce zeroes. (The proof is slightly simpler than for the full algorithm—for example, the claims of pairwise independence in Lemma 3 can be over choices of A as well as D and Y. The required condition also better matches the formal proof in [6].) Moreover, the theorem in [7] clearly doesn't contradict this proof, since the probability of an all-zeroes Hadamard transform

result on any particular pre-selected A is within the exponentially small bound allowed by the theorem. (Again, though, averaging over traced-out values of the b_i's in set A would produce a very different—and incorrect—result in the successful all-zeroes-A case, as in the previous variation.)

Finally, if we consider the original algorithm, in which set A is post-selected (randomly among sets of all-zeroes groups of samples, assuming a sufficient fixed number of groups processed), its output can easily be shown to be identical to that of the variation in which A is randomly pre-selected, conditioned on the Hadamard transforms on A in every case all producing zeroes, as hoped. The only difference is the much larger success probability, which can thus be seen to be a direct consequence of post-selecting A. It's also consistent with the theorem proven in [7] in the sense that whichever set A is ultimately post-selected, the probability that that particular set produced all zeroes when Hadamard-transformed—and the h^* and h computed from it thus deviate from the distribution dictated by the traced-out-A density matrix—was *a priori* exponentially small, and thus within the bound specified by the theorem. However, the exponentially small probabilities for each choice of A sum over the exponentially many choices of A to produce a significant (overwhelming, in fact) probability that at least one such choice is available to the algorithm, making success via post-selction possible.

To summarize, 1) the algorithm cannot work unless its output deviates non-negligibly from the traced-out-A average, but can tolerate averaging over traced-out values of the b_i's in B; 2) deviation from the traced-out-A average can occur with high probability if set A is Hadamard-transformed and the result is all zeroes; and 3) post-selection of set A allows that latter condition to hold with high probability overall, rather than an exponentially small one. Tracing out all the b_i's—including those in A—as proposed in [7], will thus produce an incorrect prediction of the algorithm's output.

Acknowledgments

The author would like to thank Sam Jaques, Sanketh Menda, Daniele Micciancio, Seyoon Ragavan, Vinod Vaikuntanathan, Thomas Vidick and Mark Zhandry for many immensely useful discussions and suggestions.

References

[1] M. Ajtai and C. Dwork. 1997. A Public-Key Cryptosystem with Worst Case/Average Case Equivalence. In *In STOC*. ACM Press, 284–293.
[2] D. Boneh and R. Lipton. 1995. Quantum cryptanalysis of hidden linear forms. In *CRYPTO*. 424–437.
[3] Z. Brakerski, E. Kirshanova, D. Stehlé, and W. Wen. 2018. Learning with Errors and Extrapolated Dihedral Cosets. In *PKC 2018*. 702–727.
[4] Y. Chen and P. Q. Nguyen. 2011. BKZ 2.0: Better lattice security estimates. In *ASIACRYPT*. 1–20.
[5] M. Ettinger and P. Høyer. 2000. On quantum algorithms for noncommutative hidden subgroups. *Adv. in Appl. Math.* 25, 3 (2000), 239–251.
[6] Y. Guo and S. Yang. 2026. Rigorous Statements and Proofs of the Lemmas in Simon's Algorithm for the Dihedral Coset Problem and Their Underlying Hypothesis. Cryptology ePrint Archive, Paper 2026/1714. https://eprint.iacr.org/2026/1714
[7] A. Gupte, S. Ragavan, and M. Zhandry. 2026. The ePrint:2026/1591 Quantum Algorithm Does Not Solve DCP. https://eprint.iacr.org/2026/1693.
[8] G. Kuperberg. 2005. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM J. on Computing* 35, 1 (2005), 170–188.
[9] D. Micciancio. [n. d.]. ([n. d.]). Personal Communication.
[10] C. Peikert. 2016. A Decade of Lattice Cryptography. *Foundations and Trends in Theoretical Computer Science* 10, 4 (2016), 283–424.
[11] S. Ragavan. [n. d.]. ([n. d.]). Personal Communication.
[12] O. Regev. 2004. Quantum Computation and Lattice Problems. *SIAM J. on Computing* 33, 3 (2004), 738–760.
[13] O. Regev. 2005. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In *In STOC*. ACM Press, 84–93.

[14] P. Shor. 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. on Computing* 26, 5 (1997), 1484–1509.