Concrete Quantum Cryptanalysis of Shortest Vector Problem

Hyunji Kim[1⋆], Kyungbae Jang[1], Anubhab Baksi[2], Sumanta Chakraborty[3], and Hwajeong Seo[1]

[1] Hansung University, Seoul, South Korea
[2] Nanyang Technological University, Singapore
[3] Techno International New Town, West Bengal, India

khj1594012@gmail.com starj1023@gmail.com anubhab001@e.ntu.edu.sg
sumantapapan@gmail.com hwajeong84@gmail.com

Abstract. This paper presents quantum circuits for the Nguyen–Vidick (NV) sieve algorithm to solve the Shortest Vector Problem (SVP) in lattice-based cryptography. We focus on optimizing the circuit depth of the quantum NV sieve, leveraging Grover's algorithm to reduce the search complexity.

Using the proposed quantum NV sieve, we estimate the quantum resources required to solve SVP for various dimensions. Specifically, for a dimension size of 512 (the parameter for Kyber-512), our implementation achieves a quantum attack cost of $2^{126.0045}$ in terms of the gate count–depth product metric used by National Institute of Standards and Technology (NIST).

To optimize circuit depth, we employ carry-lookahead and carry-save adders for efficient multi-addition operations. Further, our quantum NV sieve performs precise sieving by implementing fixed-point arithmetic, incorporating essential components (such as input setting, up-scaling, and two's complement).

To the best of our knowledge, previous work on quantum cryptanalysis of SVP using the sieve algorithm has remained theoretical, without proposing quantum circuits.

Our work humbly demonstrates that the post-quantum security of lattice-based cryptography (with respect to the quantum attack complexity) falls between that of multivariate-based and code-based cryptography.

Keywords: Shortest Vector Problem · Grover's Algorithm · Lattice-Based Cryptography · Post-Quantum Security.

1 Introduction

The rapid advancement of quantum computers poses a significant threat to modern cryptographic systems. Post-quantum cryptography (PQC) aims to address

⋆ Corresponding author

this by developing cryptosystems resistant to quantum attacks. However, quantum algorithms like Grover's search and Shor's algorithm still pose risks. For instance, Grover's search reduces the complexity of brute force attacks on block ciphers, making AES vulnerable to key search attacks [1,2,3,4,5]. On a related note, SIKE[4] was broken on a CPU for NIST security level 1 parameters [6]. Collectively, these developments underscore the need to rigorously reassess cryptographic schemes considered robust for post-quantum security.

Among PQC schemes, lattice-based cryptography is gaining increasing attention with NIST PQC finalists, such as Kyber [7], Dilithium [8], and Falcon [9] in post-quantum cryptography. Despite its promise, research shows a 45% decline in the security of lattice-based schemes since 2010[5]. Also, there are various proposals for reducing the quantum complexities [10,11,12,13,14,15] of the schemes.

Table 1: Comparison of related works for solving SVP.

Reference	Target Algorithm	Method	Device
Ishiguro et al. [16]	Sieve	Gauss Sieve	CPU
Mariano et al. [17]	Sieve	Gauss Sieve with Shared Memory	GPU
Yang et al. [18]	Sieve	Gauss Sieve with CUDA	GPU
Ducas et al [19]	Sieve	General Sieve Kernel with Tensor core	GPU
Joseph et al. [10]	Enumeration	Quantum Ising	QPU (annealer)
Bindel et al. [11]	Enumeration	BKZ	QPU (circuit)
Bai et al. [12]	Enumeration	BKZ	QPU (circuit)
Prokop et al. [13]	Enumeration	BKZ with Grover's search	QPU (circuit)
This Work	Sieve	NV sieve with Grover's search	QPU (circuit)

Efforts to solve the SVP, crucial for lattice-based cryptography, have focused on parallelizing the search for short vectors (see Table 1). Ishiguro et al. [16] implemented the Gauss Sieve [20] for a 128-dimensional lattice on CPU. The parallelized sieving on GPUs [17,18,19] have primarily targeted the core logic of the Gauss sieve. Recently, Yao Sun and Shuai Chang solved the 190-dimensional SVP Challenge [6], marking the highest dimension achieved to date. In addition to classical methods, quantum approaches to SVP have primarily focused on enumeration [10,11,12], with Prokop et al. [13] notably combining Grover's search with BKZ.

In addition, the theoretical complexity of applying the sieve algorithm on quantum computers with Grover's search has been studied [21]. To the best of our knowledge, previous works on solving the SVP using quantum computers have primarily focused on theoretical analysis rather than practical implementation. However, theoretical result is difficult to quantify the costs that may arise while implementing and simulating quantum circuits. In this work, we propose a quantum implementation of the NV sieve's core search logic with Grover's search, focusing on the efficiency and the depth optimization.

[4] An isogeny-based key encapsulation algorithm and NIST PQC finalist.
[5] https://classic.mceliece.org/comparison.html
[6] https://www.latticechallenge.org/svp-challenge/halloffame.php

1.1 Contribution

This paper presents a quantum cryptanalysis of SVP using Grover's search with a full implementation of the quantum NV sieve. Particularly, we focus on optimizing circuit depth, which is a recommended approach for Grover's algorithm (strictly speaking, for Grover's parallelization, see Appendix A). Table 2 provides a summary of our work on quantum cryptanalysis, and our contributions can be summarized as follows:

- **Depth-Optimized implementation.** To optimize circuit depth, we use Draper's quantum adder [22] (out-of-place method) for single additions (Section 3.3). Further, we employ the Quantum Carry-Save Adder (QCSA) for multi-operand additions, which is effective for handling multiple operands (Section 3.4).
- **Precise and Efficient Sieving.** In our implementation, quantum floating-point arithmetic is used to ensure precise sieving. Specifically, a fixed-point approach is adopted, taking into account computational complexity and the precision of sieving[7].
- **Essential Components for Quantum NV Sieve.** Other than that, we incorporate several non-trivial techniques for implementing essential components, such as input setting, up-scaling, and two's complement.
- **Evaluation of Post-Quantum Security.** We evaluate the post-quantum security (introduced by NIST) of lattice-based cryptography using our quantum NV sieve, and provide a comparison with the post-quantum security of code-based and MQ-based cryptography (in Section 4).

Table 2: Summary of this work with a comparison to NIST post-quantum security.

Cryptography	Method		Quantum cost	MAXDEPTH	NIST security
Code	ISD [23]	BIKE (Key)	2^{266}	✗ ($\leq 2^{40}$)	✗ ($\leq 2^{157}$)
		BIKE (Message)	2^{254}		
		HQC	2^{252}		
		McEliece	2^{266}		
Multivariate	Rank attack [24]	Rainbow (Depth opt.)	2^{93}	✓, ✗ ($\leq 2^{40}$)	✓ ($\leq 2^{157}$)
		Rainbow (Width opt.)	2^{100}		
Lattice	NV sieve (Ours)	$D = 100$	$2^{77.7615}$	✓($\leq 2^{40}$)	✓ ($\leq 2^{157}$)
		$D = 128$	$2^{83.0241}$		
		$D = 256$	$2^{101.2150}$		
		$D = 512^*$	$2^{125.8551}$		

D: Dimension of the lattice.

$*$: Corresponds to Kyber-512.

[7] If the range of the fixed-point increases, the precision of sieving improves, but the complexity also increases.

2 Preliminaries

2.1 Lattice

Lattice (L) is a set of points made up of a linear combination of basis vectors (B). Let $B = [b_1, ..., b_n] \in R^m$ be linearly independent vectors in R^m. The L generated by B is set of all the linear combinations of the column of B. The matrix B is the basis for the lattice $L(B)$. Here, dim is called the dimension of matrix B, and x is an integer. Since it is made up of lattice points, there can be more than one shortest vectors (e.g. $x, -x \in L$).

$$L(b_1, ..., b_{dim}) = \Sigma_{i=1}^{dim}(x_i \cdot b_i, x_i \in Z)$$

2.2 Lattice-based Cryptography

A single lattice can have multiple distinct bases. Although the bases differ, they generate the same lattice points. When a lattice is constructed by multiplying one basis by another, the resulting vectors form the same lattice. Bases can be categorized as good or bad. A good basis consists of short vectors, while a bad basis is obtained by multiplying a good basis by a matrix, such as an unimodular matrix [25]. Deriving a bad basis from a good one is straightforward, but extracting a good basis from a bad one is computationally difficult, making the search for short vectors essential. In lattice-based cryptography, the bad basis serves as the public key, and the good basis as the private key. Since both generate the same lattice, this design adds complexity to the decryption.

CRYSTALS-Kyber Kyber [26] is an IND-CCA2-secure key encapsulation mechanism that derives its security from the difficulty of the Learning-with-Errors (LWE) problem in module lattices. It is a finalist in the NIST post-quantum cryptography competition[8], with three parameter sets targeting different security levels (refer to Table 3).

Kyber's security is defined by three key parameters – n provides 256 bits of entropy, ensuring scalable security; k scales the lattice dimension as a multiple of n, adjusting security and efficiency (i.e., $n \cdot k$ means the lattice dimension); q is a small prime chosen for fast NTT-based multiplication and negligible failure probability for CCA security. Smaller primes do not meet security requirements.

Table 3: Parameter sets and security level for Kyber.

	n	k	q	NIST security
Kyber-512		2		≈ AES-128
Kyber-768	256	3	3329	≈ AES-192
Kyber-1024		4		≈ AES-256

[8] https://pq-crystals.org/kyber/

2.3 Shortest Vector Problem

The SVP finds the shortest nonzero vector v in a lattice L, though the solution may not be unique due to vectors of equal magnitude[9]. SVP is a fundamental problem for lattice-based cryptography, and Miklós Ajtai [27] prove that it is NP-hard. Solving SVP is especially difficult when using a bad basis, as it is unlikely to contain the shortest vector. The problem becomes even more complex as the lattice's dimensions increase.

Lattice-based cryptography algorithms typically have dimension of 500 or more, making SVP extremely challenging. To decrypt lattice-based cryptography, one must solve underlying problems such as SVP and Closest Vector Problem (CVP) [28]. In short, solving SVP poses a direct threat to the security of lattice-based cryptographic schemes.

2.4 Algorithms for Solving SVP

There are many approaches to solving lattice problems, such as enumeration and sieve algorithms. The enumeration algorithm has a super-exponential execution time, while the sieving algorithm has an exponential execution time [29].

Enumeration algorithms: These reduce the dimension of lattice [30] (e.g. Lenstra, Lenstra and Lovász (LLL) [31], Block Korkine-Zolotarev (BKZ) [32]) have been widely studied.

Sieve algorithms: The representative sieve algorithm is NV sieve [33]. The NV sieve is developed to overcome the previous impractical sieve algorithm (i.e., AKS [34]). It addresses the shortcomings of its predecessor by offering reduced time and space complexities, enhanced practicality, and actual implementation. In addition, other sieve algorithms based on NV sieve framework have been introduced, as evidenced by the studies such as Wang et al. [35], Zhang et al. [36], Laarhoven et al. [37], Becker et al. [38], and Micciancio et al. [20].

2.5 Classical NV Sieve Algorithm

Algorithm 1 briefly shows the classical NV sieve process. First, a set S is generated by randomly sampling the basis received as input. The sieve process is then performed repeatedly with S and γ as input (For the sieve process, see Figure 1). After this, the output vectors with zero vectors removed are stored in S_0, and the process is repeated until S becomes an empty set. Finally, it is completed by returning the shortest vector among the vectors belonging to S_0.

[9] SVP finds the shortest vector using the lattice vector as input, but there may be multiple vectors of equal length.

Algorithm 1: NV sieve algorithm for finding short lattice vectors

Input: A basis (B) in lattice (L), a sieve factor γ ($\frac{2}{3} < \gamma < 1$), and a number N
Output: A non-zero short vector

1: Remove all zero vectors from S. ▷ $S \leftarrow$ Sampling B using sampling algorithm
2: $S_0 \leftarrow S$
3: **Repeat**
4: $S_0 \leftarrow S$
5: $S \leftarrow$ `latticesieve`$(S, \gamma R)$ ▷ See Figure 1
6: Remove all zero vectors from S.
7: **until** S becomes an empty set.
8: **return** $v_0 \in S_0$ such that $||v_0|| = \min||v||, v \in S_0$

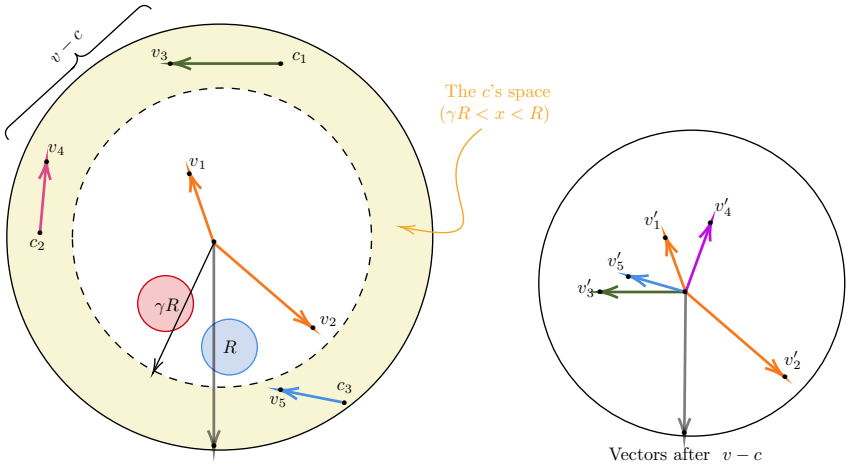


Fig. 1: Core logic in NV sieve's `latticesieve` ($\exists c \in C||v - c|| \leq \gamma R$).

The purpose of the NV sieve can be stated as follows:

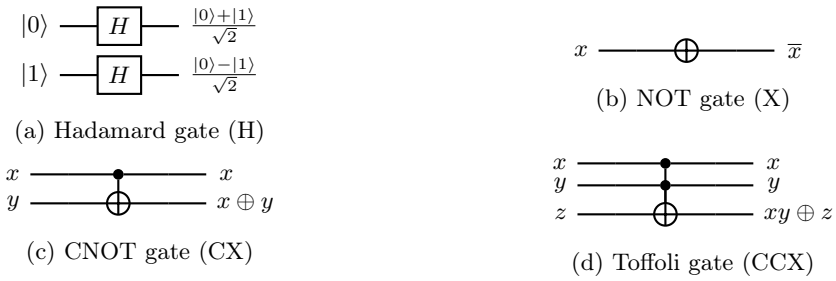
- **Reducing the search range:** When the magnitude of the longest vector is R, the search range is gradually reduced by multiplying γ (sieve factor with the range of $\frac{2}{3} < \gamma < 1$). The reduction range is determined by $\gamma \cdot R$ to obtain a shorter vector. Generally, γ is used as 0.97, and the closer γ is to 1, the more precisely the search range can be reduced.
- **Minimizing vector loss during search space reduction:** The sieve algorithm aims to reduce the search space while minimizing vector loss by applying core logic to find points that can reduce this loss. A point c on the lattice is randomly selected, and core logic is applied to c[10].

[10] c represents a sufficient number of lattice points within the range $\gamma \cdot R < x < R$.

2.6 Quantum Circuit

Qubits A qubit is the basic unit in a quantum computer and can have probabilities of 0 and 1 in the superposition state ($|\psi\rangle$). This attribute allows k qubits to represent 2^k states, and they collapse to a single classical value upon measurement.

Quantum Gates Quantum gates (see Figure 2) operate as logical gates in quantum circuits. By applying a quantum gate to a qubit, the state of the qubit can be controlled. Each gate can be used to configure superposition, entanglement, and inversion. Therefore, these gates are instrumental in computational tasks, including addition and multiplication in quantum circuits.



(a) Hadamard gate (H)

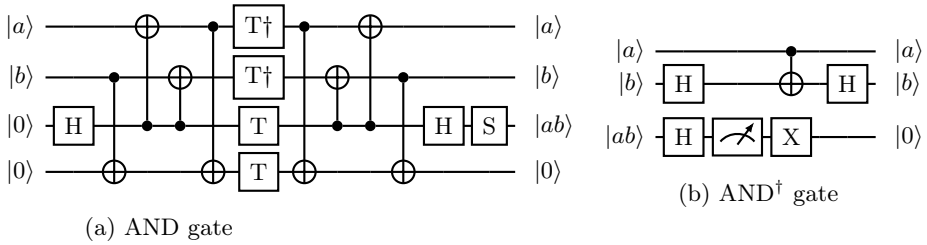
(b) NOT gate (X)

(c) CNOT gate (CX)

(d) Toffoli gate (CCX)

Fig. 2: Quantum gates.

Logical AND gate In our work, we use the logical AND gate [39] instead of the Toffoli gate to reduce the circuit depth. As shown in Figure 3, the AND gate uses 11 Clifford gates, 4 T gates, and one ancillary qubit, resulting in a T-depth of 1 and a total depth of 8. The AND† gate (the reverse of the AND gate) is built with 5 Clifford gates and 1 measurement gate, with a total depth of 4 and T-depth of 0.



(a) AND gate

(b) AND† gate

Fig. 3: Quantum AND and AND† gates

2.7 Grover's Search Algorithm

Grover's search algorithm is a quantum search algorithm for tasks with k-bit complexity and has $O(\sqrt{2^k})$ complexity ($O(2^k)$ for classical computer). The k-bit data for the target of the search must exist in a state of quantum superposition[11], given by:

$$H^{\otimes k} |0\rangle^{\otimes k} (|\psi\rangle) = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) = \frac{1}{2^{k/2}} \sum_{x=0}^{2^k-1} |x\rangle$$

Grover's search is composed of two main modules (Oracle and Diffusion):

1. *Oracle:* Oracle is a quantum circuit designed to implement the logic necessary to return a solution to the problem at hand. It achieves this by inverting the decision qubit at the circuit's conclusion as follows. The crucial aspect of Grover's search with low cost lies in the optimal implementation of the quantum circuit that constitutes the Oracle.

$$f(x) = \begin{cases} 1 \text{ if } Oracle_{\psi(k)} = Solution \\ 0 \text{ if } Oracle_{\psi(k)} \neq Solution \end{cases}$$

2. *Diffusion:* It serves to amplify the probability of the solution returned by the Oracle. By repeating this, the observation of the correct solution is increased, referred to as Grover iteration. However, it is often omitted from resource estimations [2], as its overhead is considered minimal and, therefore, negligible.

3 Quantum Circuit Implementation of NV Sieve's Core Logic for Grover's Search

In this section, we analyze and delineate the quantum implementation of the NV sieve's core logic to solve the SVP. We discuss the considerations for applying Grover's search and outline a step-by-step optimal quantum circuit for the precise oracle implementation, focusing on it. Lastly, we prove the correctness of our oracle and our approach.

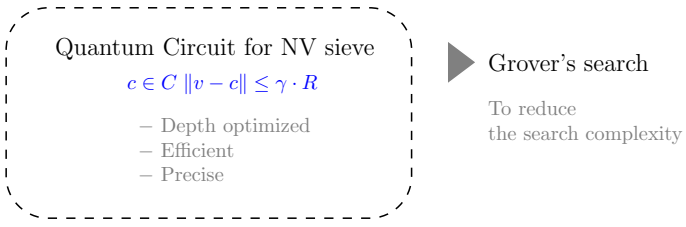


Fig. 4: Brief overview of our work.

[11] Thanks to quantum advantage, all targets are computed simultaneously.

Figure 4 illustrates a brief overview and the main contributions of our work. We focus on the quantum implementation of the sieve algorithm to solve the SVP. The core logic of the NV sieve centers around two key operations: addition and multiplication. Each operation is implemented modularly. Furthermore, Grover's search is applied to efficiently search for vector c while maintaining low complexity. For depth optimization, we employ three strategic approaches. Finally, a fixed-point implementation is introduced to ensure a precise reduction.

Here are the considerations for our work. We propose an optimized quantum implementation for the NV sieve on Grover's search. While applying Grover's search, there are two key points to focus on:

1. It is crucial to implement an accurate oracle that closely resembles the target algorithm.
2. Reducing the depth of quantum circuits is more effective than minimizing the number of qubits.

Considering the points, our design philosophy focuses on the precise implementation while striving to minimize the depth of quantum circuits.

3.1 Overview of Our Implementation for NV sieve's Core Logic

In this section, we provide an overview of the quantum implementation of the NV sieve (refer to Figure 5). Detailed descriptions of each operational module are presented in the subsequent sections.

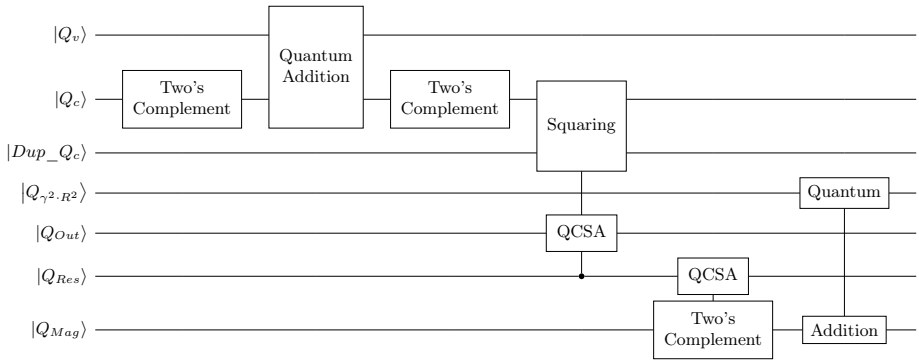


Fig. 5: An overview of the quantum circuit of NV sieve's core logic.

The main tasks for the core logic ($||v - c|| \leq \gamma R$) are the calculation of the magnitude of $v - c$ and to compare the magnitude of two vectors (γR and $||v - c||$). However, considering that we only need to compare the magnitudes, we eliminate the square root.

As shown in Algorithm 2, we need 8 steps for the oracle. In STEP 1, before implementing the core logic, we need the lattice and input setting. STEP 2

Algorithm 2: Overall steps quantum implementation for NV sieve.

Input: Reduced lattice vector (v) and sieve factor γ ($\frac{2}{3} < \gamma < 1$)

// **STEP 1: Input setting and up-scaling**
Initiate Q_v, Q_c, $Q_{\gamma^2 \cdot R^2}$ and dimension (dim).
Up-scaling(Q_v)

// **STEP 2: Two's complement on positive case**
for d in dim **do**
    Two's Complement(Q_c)
end for

// **STEP 3: Addition** $Q_v + \overline{Q_c}$ **(**$=Q_v - Q_c$**)**
for d in dim **do**
    Draper_ adder(Q_v, Q_c) ▷ See Section 3.3
end for

// **STEP 4: Two's complement on negative case and duplication**
for d in dim **do**
    Two's Complement(Q_c)
    CNOT|(Q_c, Dup_Q_c)
end for

// **STEP 5: Squaring to compute the size of vector** ▷ See Section 3.4
for d in dim **do**
    for bs in s **do**
        Q_{Res} = *Squaring* with $QCSA(Q_c, Dup\_Q_c)$
    end for
end for

// **STEP 6: Addition for squared results**
$Q_{Mag} = QCSA(Q_{Res})$

// **STEP 7: Two's complement on positive case**
Q_{Mag} =*Two's Complement*(Q_{Mag})

// **STEP 8: Size comparison between** $Q_{\gamma^2 \cdot R^2}$ **and** $(||Q_v - Q_c||)^2$
Draper_ adder($Q_{\gamma^2 \cdot R^2}$, Q_{Mag})

return $\{c_0, ..., c_{n-1}\}$

applies the two's complement to Q_c, denoted as $\overline{Q_c}$, to compute $Q_v - Q_c$. STEP 3 performs the vector subtraction as $Q_v + \overline{Q_c}$, followed by duplication Q_c in STEP 4. STEP 5 computes the square via $Q_c \times Dup\_Q_c$, and the results are summed in STEP 6 to yield the magnitude through vector addition.

In STEP 8, we calculate $\gamma^2 \cdot R^2 + \overline{Q_{Mag}}$. A resulting MSB of 0 indicates that $||v - c||$ falls within the reduced search range $\gamma \cdot R$. Therefore, the vector $v - c$ when the MSB is 0 is included in the next sieving, and the opposite case (MSB is 1) is abandoned because $v - c$ is not included in the reduced range. Ultimately, the vector Q_c that makes the vector Q_{v-c} included in the reduced search range is the solution vector of Grover's search algorithm.

3.2 Quantum Initialization and Lattice Preparation

In our quantum implementation, we employ several essential techniques:

- **Lattice and Input Setting:** Configures lattice dimensions from 10 to 512 (notably 512 for Kyber-512). And, we use 19 qubits for each vector component with a 4-qubit for fixed-point fractional precision[12] to manage overflow. We initialize Q_c applying the Hadamard gate to create superposition states for Grover's search. Additionally, the classical pre-computation of $(r \cdot R)^2$ enables an efficient magnitude comparisons without square-root operations.
- **Up-scaling:** Manages overflow of operation, which allows precise vector calculations without errors from range limitations.
- **Two's complement:** Enables accurate subtraction in vector operations and handles directionality by applying a two's complement, essential for both addition and squaring operations.

3.3 Quantum Addition for Quantum NV Sieve

Our approach prioritizes scalable, depth-optimized performance across higher dimensions, which is critical for NV sieve efficiency.

- **Subtraction via two's complement:** Subtraction is implemented as addition by performing a two's complement on the target vector. This technique is consistently used across our implementation for all general addition operations.
- **Out-of-place Draper adder:** With a bit size of $s = 19$, our implementation employs 19-qubit and 38-qubit out-of-place Draper adders (Section 4.1 in [22]) to optimize depth[13]. This design reduces T-depth compared to the in-place method but requires additional ancillary qubits.
- **Depth-Optimized Adder Design:** Given the frequent repetition of addition operations across dimensions, our design minimizes the circuit depth over the qubit count, enabling an efficient performance even in the implementations with higher dimensions.

[12] Fixed-point arithmetic is applied similarly to integer operations.

[13] The application of QCSA alongside Draper adders is further discussed in Section 3.4.

3.4 Quantum Squaring for Quantum NV Sieve

Our oracle leverages quantum squaring for vector magnitude comparisons, structured in two main phases:

Phase 1: Copying Q_c and Performing Squaring We perform squaring by multiplying Q_c and Dup_Q_c. Using CNOT with AND gates, we can save s-AND gates. In addition, we use the copied qubits, and it makes the squaring parallel (depth optimization).

Figure 6 illustrates the squaring used to obtain the magnitude of the target vector. This approach yields the output Q_{Out}, optimized for the depth while enabling a scalable parallel execution. Each component of Q_{Out} will subsequently be summed to obtain the total squared magnitude.

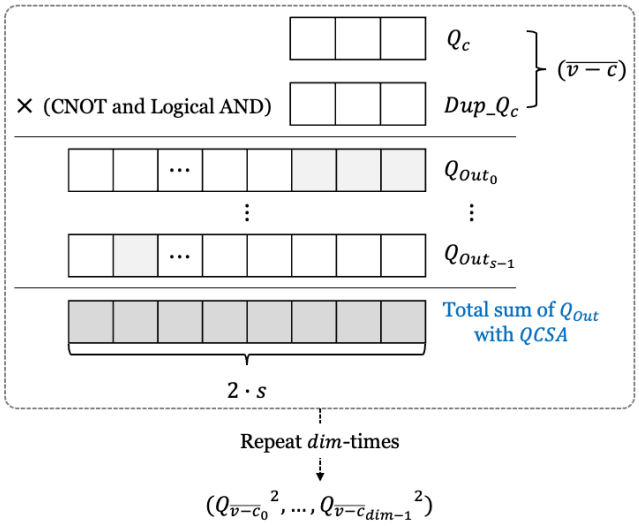


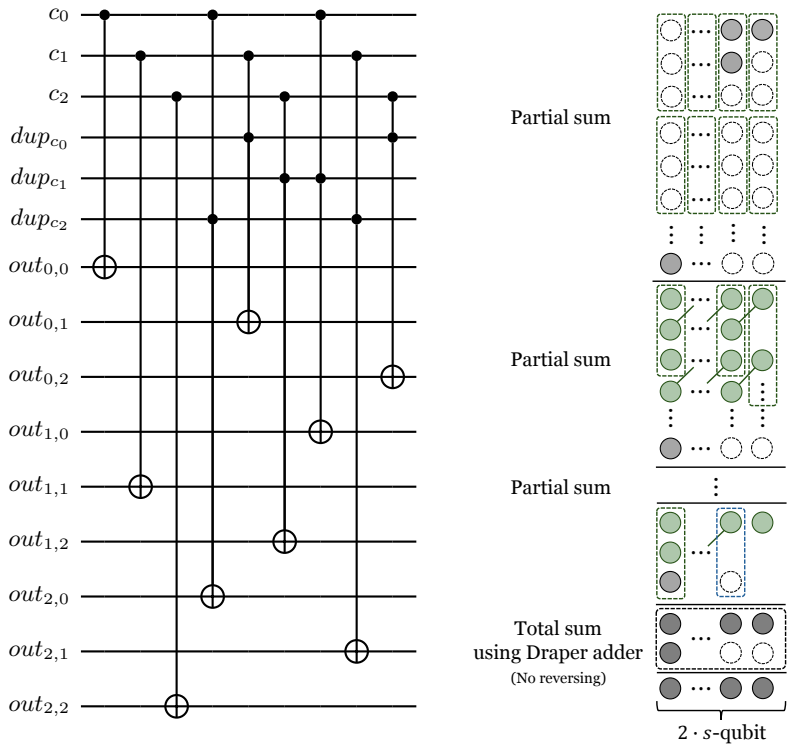
Fig. 6: Squaring using logical AND, CNOT gate and QCSA.

Phase 2: Multi-operand Addition with QCSA Our approach utilizes a QCSA [40], a multi-operand adder, to parallelize the addition. The key part requiring parallelization is the phase in which intermediate results (Q_{Out}) from the squaring process are summed simultaneously. In a general adder, the multiplication results must be added sequentially. However, the multi-operand adder increases the computational efficiency by handling multiple additions in parallel.

Figure 7b demonstrates the QCSA process. The partial sum is computed using QCSA, and then an out-of-place Draper adder is applied to complete the sum. In addition, this squaring operation is repeated n times to compute the

result for all dimensions. The result for each dimension is denoted as Q_{Res}, representing the squared components of the vector $(\overline{v-c})$:

$$Q_{Res} = ({Q_{(\overline{v-c})_0}}^2, \cdots, {Q(\overline{v-c})_{dim-1}}^2)$$



(a) Squaring (example for 3-qubit). (b) QCSA (multi-operand addition).

Fig. 7: Steps for correct squaring in quantum circuits.

3.5 Magnitude Calculation for Quantum NV Sieve

The sum of $||Q_{v-c}||^2$ represents the squared magnitude of the vector. QCSA computes it efficiently. It achieves this by summing all Q_{Res} components, as shown below:

$$Q_{Mag} = QCSA(Q_{Res}) = \sum_{i=0}^{dim-1} {Q_{(\overline{v-c})_i}}^2$$

The integration of QCSA with our squaring method enables a scalable and an efficient squared magnitude computation, significantly enhancing the quantum efficiency by minimizing the circuit depth.

4 Results and Discussion

In this section, we present the quantum resources and an in-depth analysis of our implementation. In addition, we compare the results with research that applied Grover's search to the BKZ [13]. Finally, we present a comparative analysis with other PQCs and discuss the security strength based on our experimental results.

4.1 Experiment Environment

For our experiment, we used ProjectQ, which is an open-source quantum programming tool. The target lattice dimensions range from 10 to 512. In the following sections, we denote `D` as the dimension and `FP` as the fractional parts (e.g. `D10FP4` means the lattice with dimension 10 and 4-qubit for fractional part of the fixed point.).

4.2 Results of Low Depth for Quantum NV sieve Oracle

Table 4 presents the required quantum resources for our quantum implementation. In this section, we describe results focusing on quantum depth (i.e., T-depth (Td) and Full-depth (FD)).

Table 4: Quantum resources of the quantum NV sieve's Oracle.

Case	#CNOT	#1qCliff	#T	T-depth (Td)	Full depth (FD)	Qubit (M)	Td-M	FD-M	Td^2-M	FD^2-M
D10FP4	$2^{17.3035}$	$2^{15.5008}$	$2^{15.9150}$	$2^{10.2900}$	$2^{10.9417}$	$2^{14.9981}$	$2^{25.2881}$	$2^{25.9399}$	$2^{35.5781}$	$2^{36.8817}$
D20FP4	$2^{18.2869}$	$2^{16.4790}$	$2^{16.8934}$	$2^{11.2118}$	$2^{11.7073}$	$2^{15.9827}$	$2^{27.1945}$	$2^{27.6900}$	$2^{38.4064}$	$2^{39.3974}$
D30FP4	$2^{18.8653}$	$2^{17.0554}$	$2^{17.4699}$	$2^{11.7698}$	$2^{12.2024}$	$2^{16.5613}$	$2^{28.3311}$	$2^{28.7637}$	$2^{40.1010}$	$2^{40.9661}$
D40FP4	$2^{19.2786}$	$2^{17.4680}$	$2^{17.8826}$	$2^{12.1711}$	$2^{12.5689}$	$2^{16.9749}$	$2^{29.1461}$	$2^{29.5438}$	$2^{41.3172}$	$2^{42.1127}$
D50FP4	$2^{19.5988}$	$2^{17.7876}$	$2^{18.2022}$	$2^{12.4848}$	$2^{12.8618}$	$2^{17.2952}$	$2^{29.7800}$	$2^{30.1570}$	$2^{42.2648}$	$2^{43.0189}$
D60FP4	$2^{19.8608}$	$2^{18.0492}$	$2^{18.4638}$	$2^{12.7423}$	$2^{13.1052}$	$2^{17.5572}$	$2^{30.2995}$	$2^{30.6624}$	$2^{43.0418}$	$2^{43.7677}$
D70FP4	$2^{20.0825}$	$2^{18.2707}$	$2^{18.6853}$	$2^{12.9607}$	$2^{13.3121}$	$2^{17.7789}$	$2^{30.7397}$	$2^{31.0911}$	$2^{43.7004}$	$2^{44.4033}$
D100FP4	$2^{20.5953}$	$2^{18.7829}$	$2^{19.1976}$	$2^{13.4681}$	$2^{13.7990}$	$2^{18.2918}$	$2^{31.7599}$	$2^{32.0909}$	$2^{45.2280}$	$2^{45.8899}$
D128FP4	$2^{20.9507}$	$2^{19.1382}$	$2^{19.5528}$	$2^{13.8205}$	$2^{14.1409}$	$2^{18.6473}$	$2^{32.4679}$	$2^{32.7883}$	$2^{46.2885}$	$2^{46.9293}$
D256FP4	$2^{21.9461}$	$2^{20.1366}$	$2^{20.5513}$	$2^{14.8139}$	$2^{15.1152}$	$2^{19.6462}$	$2^{34.4602}$	$2^{34.7614}$	$2^{49.2742}$	$2^{49.8766}$
D512FP4	$2^{22.9401}$	$2^{21.0885}$	$2^{21.5456}$	$2^{15.8043}$	$2^{16.0803}$	$2^{20.6463}$	$2^{36.4506}$	$2^{36.7266}$	$2^{52.2549}$	$2^{52.7059}$

Our quantum circuit for the NV sieve provides low depth since we employ 3 main strategies for depth optimization. First, we use the out-of-place Draper adder [22] in our entire quantum circuit. In-place Draper's Toffoli depth (TD) is $\lfloor s \rfloor + \lfloor s-1 \rfloor + \lfloor \frac{s}{3} \rfloor + \lfloor \frac{s-1}{3} \rfloor + 8$. However, Out-of-place Draper's Toffoli depth is $\lfloor s \rfloor + \lfloor \frac{s}{3} \rfloor + 4$. It has a lower Toffoli depth than the in-place method.

In addition, Draper adder has a lower Toffoli depth than Cuccaro [41], Takahashi [42] and Gidney [43][14].

The following equations calculate the TD for our quantum oracle. The total TD of our oracle is defined as the sum of $TD_{Draper}(Oracle)$ and $TD_{QCSA}(Oracle)$.

[14] Wang et al. [44] provides the performance comparison of the various quantum adders.

Considering FD of our oracle, we observe that TD represents approximately 80% of the overall circuit depth on average in all dimensions.

$$TD_{Draper}(Oracle) = 3 \cdot (dim \cdot TD_{Draper_s}) + 2 \cdot TD_{Draper_{2 \cdot s}} + dim$$

$$TD_{QCSA}(Oracle) = (dim + 1) \cdot TD_{QCSA_{2 \cdot s}}$$

Next, we apply the logical AND gate instead of Toffoli[15] for the computationally intensive parts. The logical AND gate requires clean qubits to store the result. In our case, we only use the out-of-place approach in our entire quantum circuit. Therefore, logical AND gate is reasonable optimization. Also, it has a T-depth of 1. In our previous work, the T-depth of the Toffoli gate we used was 4. This approach allows us to reduce the T-depth of this implementation by about a factor of four.

4.3 MAXDEPTH of Quantum NV Sieve

Our implementation indicates that the FD for 512 dimensions is approximately 2^{17}, indicating its feasibility for near-term quantum computers. The depth of 2^{40} and 2^{48} are considered reasonable MAXDEPTH [45].

Furthermore, our experimental results suggest that the increase in FD with dimension is fairly consistent. Although this work does not directly address Kyber-768 and Kyber-1024, we conservatively estimate their FD. For example, the FD for 768 dimensions is estimated to be about 1.5 times that of 512 dimensions, approximately $2^{17.7}$, and for 1024 dimensions, around $2^{18.1}$. Therefore, we expect that Kyber-768 and Kyber-1024 are unlikely to exceed the MAXDEPTH limits, indicating that no further considerations may be necessary currently.

4.4 Quantum Cost for Grover's Search

Table 5 presents the quantum cost calculated when applying Grover's search. Using AND gates keeps T_d the same as the oracle, but doubles the FD and #Total gates. In our implementation, the quantum cost for `D512FP4` is $2^{41.7494} \cdot r$, where r denotes the iteration count for Grover's search[16]. And, the appropriate r of the quantum circuit with multiple solutions (M) to the search space (N) is obtained by $r = \frac{\pi}{4} \cdot \sqrt{\frac{N}{M}}$. Here, since $M \in Z$, $\sqrt{\frac{N}{M}}$ is a smaller number than $\sqrt{N}$. Therefore, r is less than the Grover iteration with a single solution, $\frac{\pi}{4} \cdot \sqrt{N}$. Hence, a complexity of at most $O(\sqrt{N})$ can be achieved. In this analysis, we conservatively calculate the quantum cost using the maximum Grover iteration (r_{max}), which corresponds to $M = 1$ and varies depending on the dimension.

Furthermore, to calculate the final cost of the quantum NV sieve, the total number of recursive sieving iterations nv_{iter}, must be considered. Since this value

[15] This determines the T-depth of the quantum circuit. The T-depth depends on the level to which the compiler optimizes the circuit.

[16] It is a single quantum NV sieve; iteration of NV sieve is described in below.

Table 5: Quantum Cost of Grover's Search Applied to Quantum NV Sieve.

Case	#Total gates	T-depth (Td)	Full depth (FD)	Qubit (M)	Quantum cost	Td-M	FD-M	Td^2-M	FD^2-M
D10FP4	$2^{19.0421}$	$2^{10.2900}$	$2^{11.9417}$	$2^{14.9981}$	$2^{30.9839} \cdot r$	$2^{25.2881}$	$2^{26.9399}$	$2^{35.5781}$	$2^{38.8817}$
D20FP4	$2^{20.0235}$	$2^{11.2118}$	$2^{12.7073}$	$2^{15.9827}$	$2^{32.7309} \cdot r$	$2^{27.1945}$	$2^{28.6900}$	$2^{38.4064}$	$2^{41.3974}$
D30FP4	$2^{20.6011}$	$2^{11.7698}$	$2^{13.2024}$	$2^{16.5613}$	$2^{33.8036} \cdot r$	$2^{28.3311}$	$2^{29.7637}$	$2^{40.1010}$	$2^{42.9661}$
D40FP4	$2^{21.0142}$	$2^{12.1711}$	$2^{13.5689}$	$2^{16.9749}$	$2^{34.5831} \cdot r$	$2^{29.1461}$	$2^{30.5438}$	$2^{41.3172}$	$2^{44.1127}$
D50FP4	$2^{21.3342}$	$2^{12.4848}$	$2^{13.8618}$	$2^{17.2952}$	$2^{35.1960} \cdot r$	$2^{29.7800}$	$2^{31.1570}$	$2^{42.2648}$	$2^{45.0189}$
D60FP4	$2^{21.5959}$	$2^{12.7423}$	$2^{14.1052}$	$2^{17.5572}$	$2^{35.7012} \cdot r$	$2^{30.2995}$	$2^{31.6624}$	$2^{43.0418}$	$2^{45.7677}$
D70FP4	$2^{21.8176}$	$2^{12.9607}$	$2^{14.3121}$	$2^{17.7789}$	$2^{36.1297} \cdot r$	$2^{30.7397}$	$2^{32.0911}$	$2^{43.7004}$	$2^{46.4033}$
D100FP4	$2^{22.3302}$	$2^{13.4681}$	$2^{15.7990}$	$2^{18.2918}$	$2^{37.1292} \cdot r$	$2^{31.7599}$	$2^{33.0909}$	$2^{45.2280}$	$2^{47.8899}$
D128FP4	$2^{22.6855}$	$2^{13.8205}$	$2^{15.1409}$	$2^{18.6473}$	$2^{37.8265} \cdot r$	$2^{32.4679}$	$2^{33.7883}$	$2^{46.2885}$	$2^{48.9293}$
D256FP4	$2^{23.6842}$	$2^{14.8139}$	$2^{16.1152}$	$2^{19.6462}$	$2^{39.7994} \cdot r$	$2^{34.4602}$	$2^{35.7614}$	$2^{49.2742}$	$2^{51.8766}$
D512FP4	$2^{24.6691}$	$2^{15.8043}$	$2^{17.0803}$	$2^{20.6463}$	$2^{41.7494} \cdot r$	$2^{36.4506}$	$2^{37.7266}$	$2^{52.2549}$	$2^{54.7059}$

is non-trivial, it must be accounted for in the overall quantum cost calculation. The SVP challenge shows that the average length of the short vector is about 3140, which we use to estimate the recursive iterations ($iter_{nv}$) for the quantum NV sieve. About 3140 (R) is the maximum vector length in the reduced search space. Considering the 512-dimensional lattice, the initial maximum value of R_0 before reduction is around 91566. With γ set to 0.97 (commonly used in classical implementations), we recursively calculate $iter_{nv}$ for the quantum sieve as follows:

$$R_{(iter_{nv}+1)} = R_{iter_{nv}} \times \gamma$$

This means that our quantum NV sieve needs approximately $iter_{nv}$ iterations to find the short vector ($iter_{nv}$ is not r). For example, if the dimension is 512, $iter_{nv} \approx 2^{6.7909}$. In this process, by multiplying the maximum quantum cost of Grover's search (see Table 5) by $iter_{nv}$, we can determine the final quantum cost of solving SVP[17]. All $iter_{nv}$ are provided in Table 6.

Table 6 shows the maximum quantum cost with r_{max} and nv_{iter}. The quantum cost of D512FP4 is at most $2^{126.0045}$ ($=2^{41.7494} \cdot 2^{77.4642} \cdot 2^{6.7909}$)[18]. However, given that we use r_{max}, the quantum cost in actual scenario is expected to be lower than the maximum cost estimated in the presented analysis.

Moreover, Figure 8 suggest that the rate of increase of the required quantum resources decreases as the size of the lattice increases (see D10FP4, ..., D512FP4). The quantum circuits for larger lattices will obviously require more quantum resources, but we believe it will converge to trends similar to those suggested by our results. We remain some experiments on other parameters of the Kyber for future work.

[17] We apply this approach to calculate $iter_{nv}$ for all dimensions.

[18] This assumes that there is only one vector that satisfies the condition, and as the number of vectors belonging to the short vector set increases, the iteration decreases.

Table 6: Maximum quantum cost with r_{max} and nv_{iter}.

Dimension	Quantum cost w/o r	r_{max}	nv_{iter}	Maximum quantum cost with r_{max} and nv_{iter}
10	$2^{30.9839}$	$2^{10.8259}$	$2^{5.5396}$	$2^{47.3495}$
20	$2^{32.7309}$	$2^{15.3102}$	$2^{5.8553}$	$2^{53.8964}$
30	$2^{33.8036}$	$2^{18.7511}$	$2^{6.0123}$	$2^{58.5670}$
40	$2^{34.5831}$	$2^{21.6519}$	$2^{6.1142}$	$2^{62.3492}$
50	$2^{35.1960}$	$2^{24.2075}$	$2^{6.1885}$	$2^{65.5921}$
60	$2^{35.7012}$	$2^{26.5180}$	$2^{6.2465}$	$2^{68.4658}$
70	$2^{36.1297}$	$2^{28.6428}$	$2^{6.2938}$	$2^{71.0663}$
100	$2^{37.1292}$	$2^{34.2347}$	$2^{6.3976}$	$2^{77.7615}$
128	$2^{37.8265}$	$2^{38.7321}$	$2^{6.4654}$	$2^{83.0241}$
256	$2^{39.7994}$	$2^{54.7755}$	$2^{6.6401}$	$2^{101.2150}$
512 ✳	$2^{41.7494}$	$2^{77.4642}$	$2^{6.7909}$	$2^{126.0045}$

✳: Corresponds to parameter of Kyber-512.

4.5 Comparison of Quantum Resources with Related Works

Table 7 shows the quantum cost for Grover's search to solve SVP using the enumeration algorithm (BKZ, [13]) and the sieving algorithm (NV sieve, [15] and ours). It highlights the quantum cost at varying lattice dimensions, allowing for direct comparison between different implementations. Although the approaches differ, we provide a comparative analysis based on quantum cost, which serves as a fair comparison metric.

Table 7: Comparison of Quantum Cost for Grover's search with related works (our quantum cost are calculated with r_{max}).

Lattice dimension	10	20	30	40	50	70
Prokop et al. [13] (BKZ)	$2^{36.6917}$	$2^{69.2845}$	$2^{95.4466}$	$2^{172.4237}$	$2^{268.8267}$	$2^{339.5412}$
Our previous work [15] (NV sieve)	$2^{39.2568}$	$2^{51.5410}$	$2^{61.1171}$	$2^{71.9682}$	$2^{81.3414}$	$2^{99.4238}$
This work (NV sieve)	$2^{47.3495}$	$2^{53.8964}$	$2^{58.5670}$	$2^{62.3492}$	$2^{65.5921}$	$2^{71.0663}$

Kim et al. [14] is also our previous work presented at ICISC'23. However, it is excluded from Table 7 because it presented an implementation for lower dimensions. In [15] provides the quantum cost for a higher dimension and a more optimized quantum implementation than [14]. However, the Takahashi adder [46] with a higher depth is used, and the depth of the Toffoli gate used is 4. Additionally, since it does not include a fixed-point implementation, given the nature of the sieve algorithm, the implementation is less accurate in terms of lattice reduction. Compared with this work, [15] has a higher quantum cost.

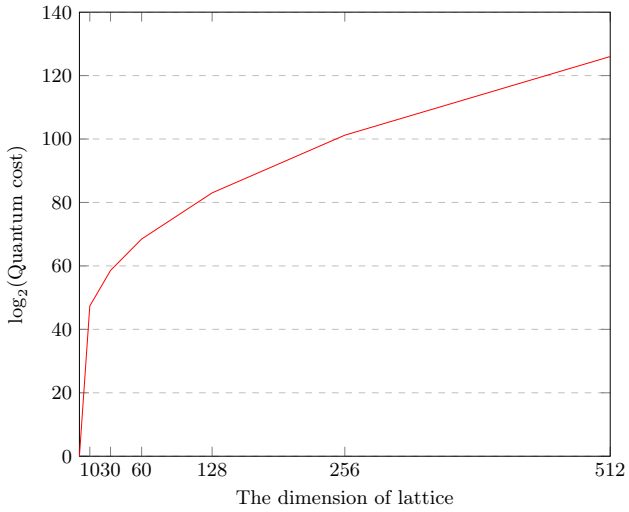


Fig. 8: The maximum quantum cost for our quantum NV sieve.

In low dimensions, similar quantum cost are required[19]. As the dimension increases, it is natural that the quantum cost increases, and an optimized implementation is necessary. Despite the increase in dimension, our approach maintains a relatively low increase in quantum cost. Since our implementation is optimized and efficient[20]. This comparison clearly demonstrates and validates the effectiveness of our approach.

4.6 Comparison with Quantum Cryptanalysis for Code/MQ-based Cryptography and Post Quantum Security

Table 8 summarizes the quantum resources required for different cryptographic schemes, including ISD (Information Set Decoding) for code-based cryptography, MinRank attack for multivariate-based cryptography, and NV sieve to solve SVP in lattice-based cryptography.

The quantum resources for ISD are based on the work by Perriello et al. [23], who evaluated several code-based algorithms including BIKE [47], HQC [48], and McEliece [49] at security levels 1, 3, and 5.

[19] In 10 and 20 dimensions, our previous work shows better performance. However, the parameters considered in this work are different from our previous work. In addition, we did not consider nv_{iter}, so it presents a lower cost.

[20] Also, the time complexity of the sieve algorithm is asymptotically faster than that of the enumeration algorithm.

Cho et al. [24] provide quantum resources for MinRank attacks on multivariate quadratic schemes such as Rainbow [50], focusing on both depth- and width-optimized quantum circuits[21].

Our implementation presents quantum resources for lattice-based cryptography (Kyber-512), corresponding to NIST security level 1. For comparison, we evaluate quantum costs and resources alongside ISD and MinRank.

Highlights of the comparisons:

- ISD for code-based cryptography requires the **highest quantum resources** (that is, FD, M, and QC) in all algorithms evaluated.
- MinRank attacks show **lower full-depth than ISD** and the **smallest qubit count** among the schemes.
- NV sieve (our work) achieves a **significantly lower full depth** than ISD and MinRank, yet incurs a higher quantum cost compared to MinRank. In terms of quantum cost, we require approximately 2^{30} times more than multivariate-based schemes and approximately 2^{170} times less than code-based schemes on average.
- In summary, our approach optimizes the FD, resulting in the lowest depth compared to other schemes, while simultaneously achieving a quantum cost that doesn't deviate significantly from NIST post-quantum security level 1.

Table 8: Comparison quantum resources between ISD for code, MinRank attack for multivariate quadratic and SVP for lattice-based cryptography.

Cryptography algorithm		Full depth (FD)	Qubit (M)	Quantum cost	FD-M	FD^2-M
ISD [23]	BIKE (key) ☆	2^{93}	2^{29}	2^{266}	2^{123}	2^{215}
	BIKE (message) ☆	2^{89}	2^{29}	2^{254}	2^{118}	2^{207}
	HQC ☆	2^{89}	2^{30}	2^{252}	2^{119}	2^{208}
	McEliece ☆	2^{92}	2^{22}	2^{266}	2^{114}	2^{206}
MinRank [24]	Rainbow ☆	2^{75}	2^{10}	2^{93}	2^{85}	2^{160}
	Rainbow ◆	2^{81}	2^{8}	2^{100}	2^{89}	2^{170}
NV sieve (Ours)	Kyber-512 ☆	2^{17}	2^{21}	2^{126}	2^{38}	2^{55}

☆/◆: Focus on depth/width optimization.

FD: NV Sieve $<$ MinRank $<$ ISD.

M, QC: MinRank $<$ NV Sieve $<$ ISD.

Therefore, we speculate that lattice-based cryptography will be more vulnerable than code-based cryptography in quantum cryptanalysis, since our im-

[21] The authors also provide circuits for MAXDEPTH, alongside the implementations mentioned in Table 8.

plementation requires fewer quantum resources than ISD for code-based cryptography. Moreover, multivariate-based cryptosystems require fewer quantum resources, so we can infer that multivariate-based cryptography is efficient but has lower quantum security than others.

Kyber-512 aims for security corresponding to NIST level 1, but considering the quantum cost of solving SVP in our implementation, it falls short of the post-quantum security level 1. Nevertheless, lattice-based cryptography continues to be a leading contender in post-quantum security with to its balance between efficiency and quantum resistance.

5 Conclusion

This work presents an efficient and precise quantum NV sieve implementation with Grover's algorithm to solve the SVP, and delivers a comprehensive analysis of quantum resource and cost in lattice dimensions of up to 512 (see Table 6).

Through the application of depth-efficient techniques, such as QCSA for multi-operand addition and out-of-place Draper adder, we achieved notably reduced depth in computation-intensive sections of the NV sieve. Our results demonstrate improvements in FD, Td, and overall quantum cost, thus reducing the resource demands for quantum attacks on lattice-based cryptosystems, specifically under Kyber-512 parameters. Importantly, our circuit implementation does not exceed the NIST MAXDEPTH for Kyber-512 and suggests that Kyber-768 and Kyber-1024 can also operate within this constraint.

Compared to ISD and the MinRank attack with Grover search, our approach yields lower FD and quantum cost, indicating a more efficient resource utilization. These findings suggest that, in terms of security, multivariate-based cryptography is less secure than lattice-based cryptography, which in turn is less secure than code-based cryptography. It can be inferred that lattice-based cryptography stands out as one of the most promising options in post-quantum security when considering the balance between efficiency and quantum resistance.

Future work will expand upon this by implementing additional parameters of Kyber and exploring the lattice-based scheme Falcon. We will target higher-dimensional lattices, applying more efficient adders, and investigating resource reuse strategies. Furthermore, we will assess the use of QRAM to achieve a speedup in quantum sieves and analyze the associated resource requirements.

A MAXDEPTH

The quantum gate count for levels is derived as the product of the gate count and the full depth (e.g., level 1 for AES-128: $2^{157} = 2^{82} \times 2^{75}$, [3, Table 11]). Additionally, NIST introduced a parameter, MAXDEPTH, to account for the extreme depth of Grover's algorithm when applied to cryptographic algorithms. If the quantum attack circuit exceeds these specified boundaries for the MAXDEPTH, it is recommended to consider parallelizing Grover's algorithm. However, Grover's

algorithm has poor performance for parallelization, as analyzed in [51,3,52]. Summarizing the analysis from [51,3,52], reducing the circuit depth by S requires increasing the number of Grover instances by S^2 (i.e., unbalanced).

If the total circuit depth D exceeds MAXDEPTH, a depth reduction using a parallel approach must be applied to satisfy MAXDEPTH. The reduction factor, S, is calculated as $\frac{D}{\text{MAXDEPTH}}$ (since D/S = MAXDEPTH). For the gate count, G, the count for each instance is reduced by S, and the number of instances increases by S^2. Thus, the estimation formula for Table **??** is derived as $G \cdot \frac{D}{\text{MAXDEPTH}}$ by $\frac{G}{S} \cdot S^2$. This formulation illustrates that NIST takes into account gate count, depth, and MAXDEPTH when estimating the complexity of quantum attacks.

In terms of the trade-off metrics, TD-M and FD-M (where M is the qubit count, TD and FD represent Toffoli and Full depths, respectively), the qubit count M is increased by S^2 (i.e., $\frac{FD^2 \cdot M}{\text{MAXDEPTH}^2}$). Thus, in parallelization, the FD-M cost changes to $\frac{FD^2 \cdot M}{\text{MAXDEPTH}}$ (with FD replaced by MAXDEPTH). In other words, the metrics of FD-M and TD-M transform into minimizing the FD^2-M and TD^2-M metrics under the constraint of MAXDEPTH.

References

1. K. Jang, S. Choi, H. Kwon, H. Kim, J. Park, and H. Seo, "Grover on Korean block ciphers," *Applied Sciences*, vol. 10, no. 18, p. 6407, 2020. 2
2. K. Jang, A. Baksi, H. Kim, G. Song, H. Seo, and A. Chattopadhyay, "Quantum analysis of AES," *Cryptology ePrint Archive*, 2022. 2, 8
3. S. Jaques, M. Naehrig, M. Roetteler, and F. Virdia, "Implementing Grover oracles for quantum key search on AES and LowMC." Cryptology ePrint Archive, Report 2019/1146, 2019. https://eprint.iacr.org/2019/1146. 2, 20, 21
4. M. Rahman and G. Paul, "Grover on katan: Quantum resource estimation," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–9, 2022. 2
5. K. Jang, S. Choi, H. Kwon, and H. Seo, "Grover on SPECK: quantum resource estimates," *Cryptology ePrint Archive*, 2020. 2
6. W. Castryck and T. Decru, "An efficient key recovery attack on sidh," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 423–447, Springer, 2023. 2
7. J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-kyber: a cca-secure module-lattice-based kem," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 353–367, IEEE, 2018. 2
8. V. Lyubashevsky, L. Ducas, E. Kiltz, T. Lepoint, P. Schwabe, G. Seiler, D. Stehlé, and S. Bai, "Crystals-dilithium," *Algorithm Specifications and Supporting Documentation*, 2020. 2
9. P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, Z. Zhang, *et al.*, "Falcon: Fast-fourier lattice-based compact signatures over ntru," *Submission to the NIST's post-quantum cryptography standardization process*, vol. 36, no. 5, pp. 1–75, 2018. 2
10. D. Joseph, A. Callison, C. Ling, and F. Mintert, "Two quantum ising algorithms for the shortest-vector problem," *Physical Review A*, vol. 103, no. 3, p. 032433, 2021. 2

11. N. Bindel, X. Bonnetain, M. Tiepelt, and F. Virdia, "Quantum lattice enumeration in limited depth," *Cryptology ePrint Archive*, 2023. 2
12. S. Bai, M.-I. van Hoof, F. B. Johnson, T. Lange, and T. Ngo, "Concrete analysis of quantum lattice enumeration," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 131–166, Springer, 2023. 2
13. M. Prokop, P. Wallden, and D. Joseph, "Grover's oracle for the shortest vector problem and its application in hybrid classical-quantum solvers," *arXiv preprint arXiv:2402.13895*, 2024. 2, 14, 17
14. H. Kim, K. Jang, Y. Oh, W. Seok, W. Lee, K. Bae, I. Sohn, and H. Seo, "Finding shortest vector using quantum NV Sieve on Grover," in *International Conference on Information Security and Cryptology*, 2023. 2, 17
15. H. Kim, K. Jang, H. Kim, A. Baksi, S. Chakraborty, and H. Seo, "Quantum nv sieve on grover for solving shortest vector problem," *Cryptology ePrint Archive*, 2024. 2, 17
16. T. Ishiguro, S. Kiyomoto, Y. Miyake, and T. Takagi, "Parallel gauss sieve algorithm: Solving the svp challenge over a 128-dimensional ideal lattice," in *Public-Key Cryptography–PKC 2014: 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings 17*, pp. 411–428, Springer, 2014. 2
17. A. Mariano, S. Timnat, and C. Bischof, "Lock-free gausssieve for linear speedups in parallel high performance svp calculation," in *2014 IEEE 26th International Symposium on Computer Architecture and High Performance Computing*, pp. 278–285, IEEE, 2014. 2
18. S.-Y. Yang, P.-C. Kuo, B.-Y. Yang, and C.-M. Cheng, "Gauss sieve algorithm on gpus," in *Cryptographers' Track at the RSA Conference*, pp. 39–57, Springer, 2017. 2
19. L. Ducas, M. Stevens, and W. van Woerden, "Advanced lattice sieving on gpus, with tensor cores," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 249–279, Springer, 2021. 2
20. D. Micciancio and P. Voulgaris, "Faster exponential time algorithms for the shortest vector problem," in *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pp. 1468–1480, SIAM, 2010. 2, 5
21. T. Laarhoven, M. Mosca, and J. Van De Pol, "Finding shortest lattice vectors faster using quantum search," *Designs, Codes and Cryptography*, vol. 77, pp. 375–400, 2015. 2
22. T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, "A logarithmic-depth quantum carry-lookahead adder," *arXiv preprint quant-ph/0406142*, 2004. 3, 11, 14
23. S. Perriello, A. Barenghi, and G. Pelosi, "Improving the efficiency of quantum circuits for information set decoding," *ACM Transactions on Quantum Computing*, vol. 4, no. 4, pp. 1–40, 2023. 3, 18, 19
24. S.-M. Cho and S.-H. Seo, "Quantum rectangular minrank attack on multi-layer uov signature schemes," *Scientific Reports*, vol. 14, no. 1, p. 16340, 2024. 3, 19
25. P. Camion, "Characterization of totally unimodular matrices," *Proceedings of the American Mathematical Society*, vol. 16, no. 5, pp. 1068–1073, 1965. 4
26. J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS – kyber: a CCA-secure module-lattice-based KEM." Cryptology ePrint Archive, Paper 2017/634, 2017. 4

27. M. Ajtai, "The shortest vector problem in L2 is NP-hard for randomized reductions," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 10–19, 1998. 5
28. D. Micciancio, "The hardness of the closest vector problem with preprocessing," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 1212–1215, 2001. 5
29. E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key {Exchange—A} new hope," in *25th USENIX Security Symposium (USENIX Security 16)*, pp. 327–343, 2016. 5
30. M. S. Esseissah, A. Bhery, S. S. Daoud, and H. M. Bahig, "Three strategies for improving shortest vector enumeration using gpus," *Scientific Programming*, vol. 2021, no. 1, p. 8852497, 2021. 5
31. P. Q. Nguyen and B. Vallée, *The LLL algorithm*. Springer, 2010. 5
32. C.-P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Mathematical programming*, vol. 66, pp. 181–199, 1994. 5
33. P. Q. Nguyen and T. Vidick, "Sieve algorithms for the shortest vector problem are practical," *Journal of Mathematical Cryptology*, vol. 2, no. 2, pp. 181–207, 2008. 5
34. M. Ajtai, R. Kumar, and D. Sivakumar, "A sieve algorithm for the shortest lattice vector problem," in *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pp. 601–610, 2001. 5
35. X. Wang, M. Liu, C. Tian, and J. Bi, "Improved nguyen-vidick heuristic sieve algorithm for shortest vector problem," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pp. 1–9, 2011. 5
36. F. Zhang, Y. Pan, and G. Hu, "A three-level sieve algorithm for the shortest vector problem," in *International Conference on Selected Areas in Cryptography*, pp. 29–47, Springer, 2013. 5
37. T. Laarhoven, "Sieving for shortest vectors in lattices using angular locality-sensitive hashing," in *Advances in Cryptology–CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I 35*, pp. 3–22, Springer, 2015. 5
38. A. Becker, N. Gama, and A. Joux, "Speeding-up lattice sieving without increasing the memory, using sub-quadratic nearest neighbor search," *Cryptology ePrint Archive*, 2015. 5
39. S. Jaques, M. Naehrig, M. Roetteler, and F. Virdia, "Implementing grover oracles for quantum key search on aes and lowmc," in *Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part II 30*, pp. 280–310, Springer, 2020. 7
40. P. Gossett, "Quantum carry-save arithmetic," *arXiv preprint quant-ph/9808061*, 1998. 12
41. S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, "A new quantum ripple-carry addition circuit," *arXiv preprint quant-ph/0410184*, 2004. 14
42. Y. Takahashi and N. Kunihiro, "A fast quantum circuit for addition with few qubits," *Quantum Information & Computation*, vol. 8, no. 6, pp. 636–649, 2008. 14
43. C. Gidney and M. Ekerå, "How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits," *Quantum*, vol. 5, p. 433, 2021. 14
44. S. Wang, S. Deb, A. Mondal, and A. Chattopadhyay, "Optimal toffoli-depth quantum adder," *arXiv preprint arXiv:2405.02523*, 2024. 14
45. C. NIST, "Submission requirements and evaluation criteria for the post-quantum cryptography standardization process," 2016. 15

46. Y. Takahashi, S. Tani, and N. Kunihiro, "Quantum addition circuits and unbounded fan-out," *arXiv preprint arXiv:0910.2530*, 2009. 17
47. N. Aragon, P. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Ghosh, S. Gueron, T. Güneysu, *et al.*, "Bike: bit flipping key encapsulation," *NIST Post-Quantum Cryptography Standardization Process*, 2022. 18
48. C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, G. Zémor, and I. Bourges, "Hamming quasi-cyclic (hqc)," *NIST PQC Round*, vol. 2, no. 4, p. 13, 2018. 18
49. R. J. McEliece, "A public-key cryptosystem based on algebraic," *Coding Thv*, vol. 4244, pp. 114–116, 1978. 18
50. J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *International conference on applied cryptography and network security*, pp. 164–175, Springer, 2005. 19
51. P. Kim, D. Han, and K. C. Jeong, "Time–space complexity of quantum search algorithms in symmetric cryptanalysis: applying to AES and SHA-2," *Quantum Information Processing*, vol. 17, pp. 1–39, 2018. 21
52. D. Sarah and C. Peter, "On the practical cost of grover for aes key recovery," 2024. https://csrc.nist.gov/csrc/media/Events/2024/fifth-pqc-standardization-conference/documents/papers/on-practical-cost-of-grover.pdf. 21