Quantum Walks for Collision-Based Information Set Decoding

Simone Perriello
Department of Electronics, Information and Bioengineering (DEIB)
Politecnico di Milano
Milano, Italy
simone.perriello@polimi.it

Alessandro Barenghi
Department of Electronics, Information and Bioengineering (DEIB)
Politecnico di Milano
Milano, Italy
alessandro.barenghi@polimi.it

Gerardo Pelosi
Department of Electronics, Information and Bioengineering (DEIB)
Politecnico di Milano
Milano, Italy
gerardo.pelosi@polimi.it

Abstract

Code-based cryptography is central for post-quantum cryptography, with security grounded in the hardness of the Syndrome Decoding Problem (SDP). The most effective attacks are variants of Information Set Decoding (ISD), which remain exponential-time even in the quantum setting, where known techniques provide at most quadratic speedups via quantum amplitude amplification (QAA). In this work, we present the first gate-level realization of quantum walks (QW) for Stern-like collision-based ISD, and present a novel circuit for the update operator of the QW, correcting issues in prior proposals, while avoiding the use of exponential quantum random-access memory. Our construction enables a circuit-level complexity analysis under realistic cost models and allows a direct comparison between QW-based and QAA-based ISD. We evaluate our approach on the cryptographic schemes that reached the final stages of international standardization. Despite the richer algorithmic structure of QW, we show that they do not outperform plain QAA-based ISD for practical instances of the SDP, as the Gauss–Jordan elimination subroutine dominates the overall cost. More generally, we provide circuit-level evidence that QW offer an advantage over QAA only when the oracle cost is asymptotically smaller than the cost of the QW operations themselves. These results clarify the practical algorithmic limits of QW techniques and contribute to a more accurate assessment of quantum speedups for structured search problems in cryptanalysis.

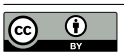
CCS Concepts

• **Theory of computation** → **Design and analysis of algorithms**; • **Mathematics of computing** → *Graph algorithms*; • **Computer systems organization** → **Quantum computing**.

Keywords

Quantum walks, Quantum amplitude amplification, Information Set Decoding

ACM Reference Format:
Simone Perriello, Alessandro Barenghi, and Gerardo Pelosi. 2026. Quantum Walks for Collision-Based Information Set Decoding. In *Proceedings of the 23rd ACM International Conference on Computing Frontiers (CF '26), May 19–21, 2026, Catania, Italy*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3801487.3801826



CF '26, Catania, Italy

ACM ISBN 979-8-4007-2568-5/26/05
https://doi.org/10.1145/3801487.3801826

1 Introduction

Shor's quantum algorithm [34] shows that integer factorization and discrete logarithms can be solved in polynomial time on a quantum computer, invalidating the hardness assumptions underlying widely used systems such as RSA and elliptic-curve cryptography. This breakthrough has triggered a large-scale transition toward cryptographic constructions believed to remain secure against quantum adversaries. Under the umbrella of *post-quantum cryptography (PQC)*, standardization efforts led by the *National Institute of Standards and Technology (NIST)* aim to identify and deploy such alternatives. Among the proposed PQC families, code-based cryptography stands out as one of the most mature and conservative approaches. Its security is rooted in the hardness of the *Syndrome Decoding Problem (SDP)*, whose decision version is NP-complete [4]. This confidence is reflected in NIST's 2025 standardization of the code-based key-encapsulation mechanism HQC [1], as well as in the continued presence of code-based proposals in NIST's calls for post-quantum digital signatures [27]. As a result, accurately quantifying the resources required to attack generic code-based schemes is crucial for sound parameter selection and security assessment.

From a classical perspective, the most effective attacks against generic code-based constructions belong to the family of *Information Set Decoding (ISD)* algorithms. Since Prange's original proposal [32], ISD has undergone a long sequence of refinements [3, 6, 9, 10, 12, 21, 22, 24, 25, 35], reducing asymptotic running times while remaining exponential-time.

Related works. The first quantum adaptation of ISD was proposed in [5], where Grover-style QAA [7, 11] yields a quadratic asymptotic speedup over classical search. Subsequent works moved beyond asymptotic analyses and developed explicit gate-level realizations [29–31]. These approaches rely on two core ingredients: deterministic Dicke-state preparation circuits [2] to generate uniform superpositions over combinatorial domains, and a reversible implementation of Gauss–Jordan elimination, known as *quantum Gauss–Jordan elimination (QGJE)*.

A different line of research focuses on collision-based ISD variants and employs *quantum walks (QWs)* on structured graphs. In particular, theoretical works such as [15, 17] model the combinatorial structure of algorithms like Stern's decoding via QWs on Cartesian products of Johnson graphs. However, these approaches rely on an exponential amount of *quantum random-access memory (QRAM)*, whose physical realizability remains uncertain [13]. At

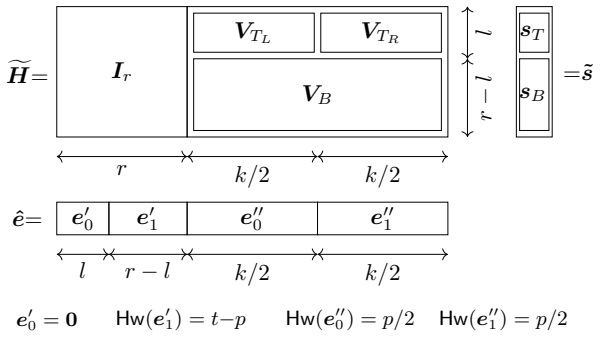


Figure 1: Weight distribution assumed in Stern algorithm.

the same time, no gate-based implementation of such approaches has ever been shown.

Contributions. We present a complete gate-level realization of a QW tailored to a Stern-like decoding strategy, fully adapted to the quantum setting. To the best of our knowledge, this is the first such construction. Our proposal avoids the use of exponential QRAM, relying instead on explicit, fully reversible subcircuits. Our framework enables a direct comparison between QW-based ISD and QAA-based approaches. We show that, despite the richer walk structure, the resulting concrete complexities do not outperform those of simpler ISD variants such as Prange or Lee–Brickell. The underlying reason is that the QGJE subroutine remains the dominant cost, acting as a bottleneck in both settings.

These results provide circuit-level evidence that QWs can outperform plain QAA-based ISD only when the oracle complexity is sufficiently smaller than the cost of the walk operations themselves. In doing so, our work clarifies the practical limits of QW techniques for generic cryptanalytic problems and contributes to a more accurate assessment of the security margins of post-quantum cryptosystems.

2 Background

Notation. Calligraphic uppercase letters (e.g., $\mathcal{S}$) denote sets, whose cardinality is written as $|\mathcal{S}|$. The notation $\{x\}$ denotes the set $\{0,\dots,x-1\}$, and write $\binom{\{x\}}{y}$ for the collection of all y-element subsets of $\{x\}$. Unless otherwise stated, vectors are column vectors and are denoted by lowercase bold letters (e.g., $\boldsymbol{v}$), while matrices are denoted by uppercase bold letters (e.g., $\boldsymbol{M}$). The notation $\boldsymbol{v}[i]$ refers to the i-th entry of a vector $\boldsymbol{v}$, whereas $\boldsymbol{M}[i]$ denotes the i-th column of a matrix $\boldsymbol{M}$. Data structures such as (associative) arrays are represented using sans-serif symbols (e.g., D), with $\mathsf{D}[k]$ denoting access to the value associated with the key k.

2.1 Collision-based Information Set Decoding

Code-based cryptography comprises public-key cryptographic schemes whose security relies on the presumed hardness of decoding random linear codes. The central computational problem underlying these constructions is the *Syndrome Decoding Problem (SDP)*: given a parity-check matrix $\boldsymbol{H} \in \mathbb{F}_2^{r\times n}$, a syndrome $\boldsymbol{s} \in \mathbb{F}_2^r$, and a weight bound $t \in \mathbb{N}$, find a vector $\boldsymbol{e} \in \mathbb{F}_2^n$ of Hamming weight at most t such that $\boldsymbol{He} = \boldsymbol{s}$. This corresponds to an underdetermined linear system of $r = n - k$ equations in n variables, where uniqueness of the solution is enforced by the Hamming-weight constraint. The SDP is NP-hard and is widely believed to remain intractable even in the presence of quantum computers.

Algorithm 1: Stern Algorithm

Input : $\boldsymbol{H}$: a $r \times n$ binary parity-check matrix
$\boldsymbol{s}$: a r-bit long syndrome (column vector)
t: the weight of the error vector to be recovered
p: the weight of the first k bits of $\hat{\boldsymbol{e}}$, $0 \le p \le t$
Output : $\boldsymbol{e}$: $n \times 1$ column vector s.t. $\boldsymbol{He} = \boldsymbol{s}$, $\mathsf{Hw}(\boldsymbol{e}) = t$

1 **repeat**
2 **repeat**
3 $\mathcal{I} \xleftarrow{\$} \binom{\{n\}}{k}$
4 $\boldsymbol{P} \leftarrow \mathsf{GetPermutation}(\mathcal{I})$
5 $\widehat{\boldsymbol{H}} \leftarrow \boldsymbol{HP}$ // implies $\hat{\boldsymbol{e}}=\boldsymbol{P}^T\boldsymbol{e}$
6 $\left[\widetilde{\boldsymbol{H}} \,\middle|\, \tilde{\mathsf{s}}\right] \leftarrow \mathsf{GJE}\left(\left[\widehat{\boldsymbol{H}} \,\middle|\, \boldsymbol{s}\right]\right)$
7 $[\boldsymbol{W}_{r\times r} \mid \boldsymbol{V}_{r\times k}] \leftarrow \widetilde{\boldsymbol{H}}$
8 **until** $\boldsymbol{W} = \boldsymbol{I}_r$
9 **foreach** $\mathcal{J}_0 \in \binom{\{\lfloor k/2-1\rfloor\}}{\lfloor p/2\rfloor}$ **do**
10 $\boldsymbol{s}_{T_0} \leftarrow \boldsymbol{s}_T \oplus \left(\bigoplus_{j\in\mathcal{J}_0} \boldsymbol{V}_{T_L}[j]\right)$
11 $\mathsf{D}[\boldsymbol{s}_{T_0}] \leftarrow \mathcal{J}_0$
12 **foreach** $\mathcal{J}_1 \in \binom{\{\lceil k/2-1\rceil\}}{\lceil p/2\rceil}$ **do**
13 $\boldsymbol{s}_{T_1} \leftarrow \bigoplus_{j\in\mathcal{J}_1} \boldsymbol{V}_{T_R}[j]$
14 **if** $\mathsf{D}[\boldsymbol{s}_{T_1}] \neq \emptyset$ **then**
15 $\mathcal{J} \leftarrow \mathcal{J}_0 \cup \{j + \lfloor k/2\rfloor \mid j \in \mathcal{J}_1\}$
16 $\boldsymbol{e}'_1 \leftarrow \boldsymbol{s}_B \oplus \left(\bigoplus_{j\in\mathcal{J}} \boldsymbol{V}_B[j]\right)$
17 **if** $\mathsf{Hw}(\boldsymbol{e}'_1) = t - p$ **then**
18 $\boldsymbol{e}'' \leftarrow \mathbf{0}_k$
19 **for** $j \in \mathcal{J}$ **do** $\boldsymbol{e}''_j \leftarrow 1$
20 $\hat{\boldsymbol{e}} \leftarrow \left[\mathbf{0}_l, \boldsymbol{e}'_1, \boldsymbol{e}''\right]$
21 **return** $\boldsymbol{P}\hat{\boldsymbol{e}}$

The primary cryptanalytic approach against generic code-based schemes is *Information Set Decoding (ISD)*, a class of Las Vegas algorithms that aim to recover the error vector by repeatedly guessing subsets of code positions. Since its original introduction by Prange [32], ISD has undergone numerous refinements [21, 22, 35], which substantially improve practical performance while preserving exponential asymptotic complexity. More recently, quantum algorithms have been shown to further reduce the cost of ISD attacks. Early works focused on asymptotic analyses without addressing circuit-level realizations [15, 17], while subsequent contributions provided explicit implementations in the quantum circuit model, enabling concrete resource estimates in terms of gate counts, depth, and qubit usage [8, 29, 30].

Among classical ISD variants, Stern's algorithm [35] is a seminal example of a *collision-based* decoding strategy. Building on the original Prange framework, Stern's method employs a meet-in-the-middle approach to reduce the time complexity of solving the SDP, at the cost of an exponential memory overhead. The algorithm, summarized in Alg. 1, begins by selecting uniformly at random an *information set* $\mathcal{I} \in \binom{\{n\}}{k}$ (line 3), which represents a guess on the k positions of the error vector $\boldsymbol{e}$ containing exactly p nonzero entries.

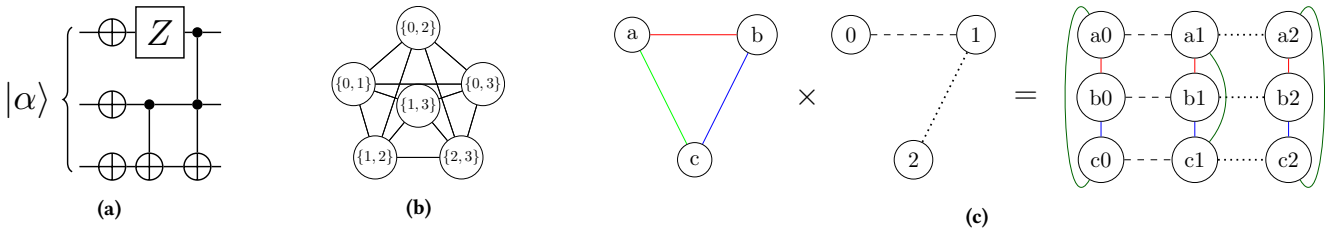


Figure 2: Examples of (a) a quantum circuit, (b) the Johnson graph $J(4,2)$, and (c) a Cartesian product of graphs.

The complementary index set $\mathcal{I}^* = \{n\} \setminus \mathcal{I}$ is then assumed to contain the remaining $t-p$ nonzero entries of $\boldsymbol{e}$.

The indices in $\mathcal{I}$ define a permutation matrix $\boldsymbol{P}$ (line 4), which rearranges the columns of the parity-check matrix $\boldsymbol{H}$ so that those indexed by $\mathcal{I}$ are moved to the rightmost positions (line 5), yielding the permuted matrix $\widehat{\boldsymbol{H}}$. Applying the same permutation to the error vector produces the equivalent syndrome equation $\boldsymbol{s} = \widehat{\boldsymbol{H}}\hat{\boldsymbol{e}}$, where $\hat{\boldsymbol{e}} = \boldsymbol{P}^T\boldsymbol{e}$. To solve the previous system of equations, the algorithm applies a *Gauss–Jordan elimination (GJE)* to the augmented matrix obtained by concatenating $\widehat{\boldsymbol{H}}$ and $\boldsymbol{s}$ (line 6). The goal is to transform the leftmost part of the resulting matrix $\widetilde{\boldsymbol{H}}$ into the identity matrix $\boldsymbol{I}_r$, a step that fails with probability approximately 0.288 [29].

Conditioned on the success of the previous step, Stern's algorithm introduces additional structural assumptions to enable a collision search. As illustrated in Fig. 1, the portion of $\hat{\boldsymbol{e}}$ indexed by $\mathcal{I}$, denoted $\boldsymbol{e''}$, is split into two halves of size $k/2$, each assumed to have Hamming weight $p/2$. Simultaneously, the portion indexed by $\mathcal{I}^*$, denoted $\boldsymbol{e'}$, is assumed to contain a run of l zero entries in its leading coordinates, while the remaining $r-l$ coordinates have Hamming weight $t-p$. Accordingly, the permuted parity-check matrix is partitioned into a top block of l rows and a bottom block of $r-l$ rows, with the top block further divided into two submatrices, $\boldsymbol{V}_{T_L}$ and $\boldsymbol{V}_{T_R}$. Under these assumptions, the top l coordinates of the syndrome equation reduce to

$$\boldsymbol{s}_T = \bigoplus_{j\in\mathcal{J}_0} \boldsymbol{V}_{T_L}[j] \oplus \bigoplus_{j\in\mathcal{J}_1} \boldsymbol{V}_{T_R}[j] \rightarrow$$
$$\boldsymbol{s}_T \oplus \bigoplus_{j\in\mathcal{J}_0} \boldsymbol{V}_{T_L}[j] = \bigoplus_{j\in\mathcal{J}_1} \boldsymbol{V}_{T_R}[j] \tag{1}$$

where $\mathcal{J}_0$, $\mathcal{J}_1$ index the columns selected in the two halves of $\boldsymbol{e''}$.

To solve Eq. 1, Stern's algorithm performs a collision search between the two halves. One side of the equation is exhaustively computed for all choices of $p/2$ indices and stored in an associative data structure (lines 9–11). The other side is then computed on the fly (lines 12–13) and queried against this structure (line 14). Each detected collision produces a candidate partial solution identifying p error positions out of the total weight t. This candidate is then validated by verifying that the remaining coordinates satisfy the required Hamming weight constraint $t-p$ (lines 15–21).

Being a Las Vegas algorithm, the expected running time T of Stern's algorithm is given by the ratio between the cost of a single iteration T_{iter} and the success probability per iteration P_{iter}, namely $T = \frac{T_{\text{iter}}}{P_{\text{iter}}}$. The success probability corresponds to the fraction of error vectors satisfying Stern's constraints, and is given by

$$P_{\text{iter}} = \binom{r-l}{t-p}\binom{k/2}{p/2}^2 / \binom{n}{t}. \tag{2}$$

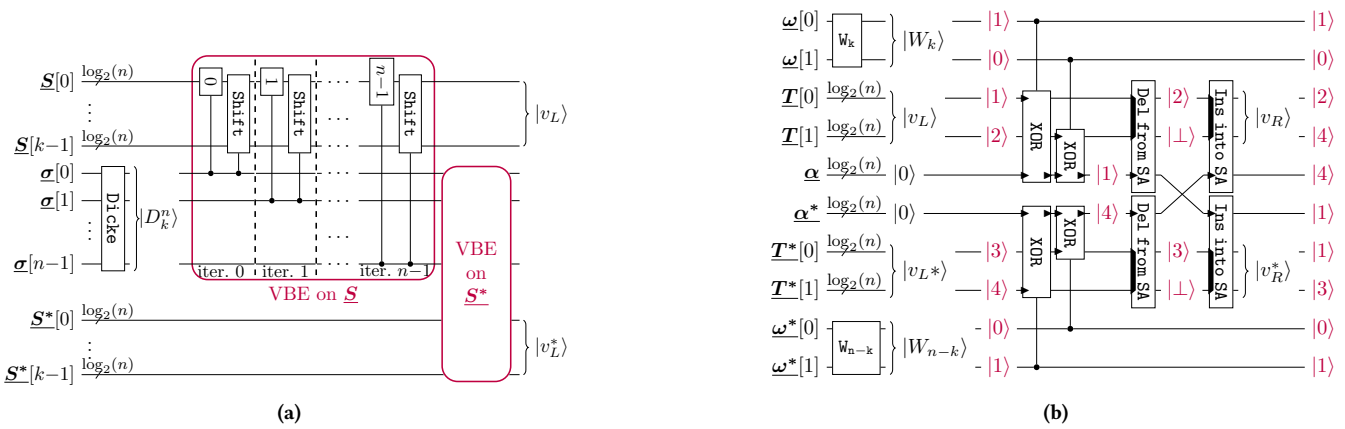
The cost of a single iteration can be decomposed into three dominant contributions. First, the algorithm extracts an information set and performs GJE, whose cost is $\in O(n^3)$. Second, D is populated by summing, for each of the $\binom{k/2}{p/2}$ choices in $\mathcal{J}_0$, $\frac{p}{2}$ columns of $\boldsymbol{V}$ of length l, yielding a total cost $\in O\left(\binom{k/2}{p/2}\frac{p}{2}\,l\right)$. Third, for each of the $\binom{k/2}{p/2}$ candidates in $\mathcal{J}_1$, the partial sum $\boldsymbol{s}_{T_1}$ is computed at the same asymptotic cost. When a collision is detected, an additional sum of p vectors of length $r-l$ is performed at cost $\in O(p(r-l))$; this event occurs with probability $\binom{k/2}{p/2}/2^l$. The space complexity is dominated by the storage required for the associative data structure, which contains $\binom{k/2}{p/2}$ entries. Each entry stores an l-bit vector $\boldsymbol{s}_{T_0}$ together with the corresponding $\frac{k}{2}$ indices defining the set $\mathcal{J}_0$.

2.2 Quantum Computing

Quantum algorithms operate on *qubits*, which are vectors in the two-dimensional complex Hilbert space $\mathbb{C}^2$. Using Dirac notation, a qubit state is written as $|\psi\rangle = a_0|0\rangle + a_1|1\rangle$, where $a_0, a_1 \in \mathbb{C}$ and $|a_0|^2 + |a_1|^2 = 1$. An n-qubit system is described by a vector in a 2^n-dimensional Hilbert space, $|\psi\rangle = \sum_{i\in\{0,1\}^n} a_i|i\rangle$. A measurement returns a basis state $|i\rangle$ with probability $|a_i|^2$. The evolution of a quantum system is governed by *unitary operators*, i.e., matrices $\boldsymbol{U} \in \mathbb{C}^{2^n\times 2^n}$ satisfying $\boldsymbol{U}\boldsymbol{U}^\dagger = I$. Applying $\boldsymbol{U}$ to a state $|\psi\rangle$ produces the new state $\boldsymbol{U}|\psi\rangle$. Sequences of unitary operators are applied from right to left, as in $\boldsymbol{U}_d \cdots \boldsymbol{U}_1|\psi\rangle$.

Quantum algorithms are typically described in the *circuit model*, where unitary transformations are represented as *quantum gates*. This work relies on a standard set of elementary gates, illustrated in Fig. 2(a). The NOT gate flips a qubit between the computational basis states $|0\rangle$ and $|1\rangle$. The controlled-NOT (CNOT) gate flips a target qubit conditioned on the control being in state $|1\rangle$, implementing the classical XOR operation, while the CCNOT gate generalizes this mechanism to two control qubits, implementing the classical AND operation. The Z gate applies a phase flip to the $|1\rangle$ state. We adopt the hybrid model of [14], in which a classical controller orchestrates quantum routines acting on *quantum registers*. Quantum registers are denoted by underlined symbols (e.g., $\underline{v}$).

Reflection operators. *Reflection operators* are unitary transformations that reflect a quantum state with respect to a given subspace. We denote by $\boldsymbol{R}_\alpha$ the reflection about the one-dimensional subspace



(a) (b)

Figure 3: (a) Gate-level realization of the operator $U_{\mathcal{V}}$ proposed in [20], composed of a Dicke state preparation circuit followed by a *Vertex Binary Encoder (VBE)*. (b) Gate-based implementation of the update operator U_u introduced in [20], shown on a toy instance with n=4 and k=2. Left/right triangles denote input/output quantum registers; trapezoids form a *Sorted Array (SA)*.

spanned by the state $|\alpha\rangle$. Its action is defined by $\boldsymbol{R}_\alpha |\alpha\rangle = |\alpha\rangle$ and $\boldsymbol{R}_\alpha |\alpha^\perp\rangle = -|\alpha^\perp\rangle$, where $|\alpha^\perp\rangle$ denotes any state orthogonal to $|\alpha\rangle$. Similarly, $\boldsymbol{R}_{\alpha^\perp}$ denotes the reflection about the subspace orthogonal to $|\alpha\rangle$. Two such operators are of particular relevance in the circuit model considered in this work. The operator $\boldsymbol{R}_{1^\perp}$ flips the sign of the amplitude associated with the computational basis state $|1^m\rangle$. It can be implemented as a $\mathtt{C}^m\mathtt{Z}$ gate, in which all m qubits act as controls for a multi-controlled Z operation targeting $|1^m\rangle$. Analogously, the operator $\boldsymbol{R}_{0^\perp}$ flips the sign of the amplitude associated with the basis state $|0^m\rangle$. Its implementation consists of applying a layer of NOT gates to all m qubits, followed by a $\mathtt{C}^m\mathtt{Z}$ gate, and a final layer of NOT gates to restore the computational basis.

2.3 Quantum Walk search on Johnson graphs

Quantum-walk (QW)-based search can be viewed as a generalization of tre *Quantum Amplitude Amplification (QAA)* technique [7, 11], in which the unstructured uniform search space is replaced by the vertex set of a graph encoding additional combinatorial structure.

Let G=($\mathcal{V}, \mathcal{E}$) be an undirected graph whose vertices represent candidate solutions, and let $f : \mathcal{V} \to \{0, 1\}$ be a Boolean predicate defining the marked set $\mathcal{M}$=$\{v \in \mathcal{V} \mid f(v)=1\}$. Random-walk-based search algorithms move locally between adjacent vertices according to a transition matrix $\boldsymbol{P}$ and rely on the mixing properties of the resulting Markov chain. In particular, for an ergodic and reversible chain, the walk converges to a unique stationary distribution $\boldsymbol{s}_\pi$ at a rate governed by the reciprocal of the spectral gap δ of $\boldsymbol{P}$. Using a uniform Markov chain (i.e., one in which the stationary distribution corresponds to the uniform distribution) the search algorithm begins with a uniform distribution $\boldsymbol{s}_\pi$ over the vertices. The algorithm samples one element v_i from it and checks if $f(v_i)$=1. If not, the algorithm performs $1/\delta$ steps of the random walk to reestablish the uniform distribution and repeats the process.

In the quantum setting, this paradigm is captured by the MNRS framework introduced in [23]. The framework achieves the same quadratic speedup in the fraction of marked elements as QAA, while exploiting the locality of the underlying graph to obtain improved concrete performance in structured problems. A central role in these constructions is played by *Johnson graphs* $J(n, k)$. These graphs are regular and undirected, with vertices corresponding to all k-subsets of $\{n\}$, and edges connecting pairs of subsets that differ in exactly one element. Johnson graphs have $|\mathcal{V}| = \binom{n}{k}$ vertices, degree $k(n-k)$, and spectral gap $\delta = \Theta(1/k)$ [33], making them well suited for QW search. An example is shown in Fig. 2(b).

The MNRS search algorithm is organized around three main components, which we summarize below.

1) Input preparation. While QAA initializes a uniform superposition of domain elements $\boldsymbol{U}_{\mathcal{V}}$, the MNRS framework initializes a uniform superposition over edges of the graph through the operator $\boldsymbol{U}_{\mathcal{E}}$. Concretely, the prepared input state is

$$|\epsilon\rangle = \frac{1}{\sqrt{|\mathcal{E}|}} \sum_{(v_L, v_R) \in \mathcal{E}} |v_L\rangle\,|v_R\rangle\,, \tag{3}$$

where v_L and v_R denote the source and destination vertices of an edge, respectively.

2) Oracle. The oracle acts on the register encoding the source vertex v_L and flips the phase of all basis states corresponding to vertices $\in \mathcal{M}$. Given a reversible implementation of the Boolean function f, named $\boldsymbol{U}_f$, the oracle operator can be implemented as $\boldsymbol{R}_{\mathcal{M}^\perp} = \boldsymbol{U}_f \boldsymbol{R}_{0^\perp} \boldsymbol{U}_f^\dagger$.

3) Diffusion. While QAA implements the diffusion operator as the sequence $\boldsymbol{U}_{\mathcal{V}}^\dagger \boldsymbol{R}_{0^\perp} \boldsymbol{U}_{\mathcal{V}}$, the MNRS framework replaces this step with a *quantum phase estimation (QPE)* on the *walk operator* $\boldsymbol{U}_w$, the latter providing a unitary implementation of the classical transition matrix $\boldsymbol{P}$. In the construction of [23], the walk operator is realized as $\boldsymbol{U}_w = \boldsymbol{U}_u\, \boldsymbol{R}_{0^\perp}\, \boldsymbol{U}_u^\dagger$, and is applied separately to the registers encoding the source and destination vertices. The QPE returns, on a separate register, the estimated phase of $\boldsymbol{U}_w$: a phase of 0, corresponding to the eigenvalue 1, identifies the uniform stationary distribution; in such a case, the state is flipped using a $\boldsymbol{R}_{0^\perp}$ operator. This procedure requires $\Theta(1/\sqrt{\delta})$ controlled applications of $\boldsymbol{U}_w$.

Cartesian product of Johnson graphs. Given two graphs $G_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $G_2 = (\mathcal{V}_2, \mathcal{E}_2)$, their cartesian product has vertex set $\mathcal{V}_1 \times \mathcal{V}_2$, whose elements represent ordered pairs of vertices from

the two factor graphs. fwo vertices (v_{1L}, v_{1R}) and (v_{2L}, v_{2R}) are adjacent if they differ in exactly one coordinate and that difference corresponds to an edge in the associated factor graph. As a result, a single step of a walk on the product graph updates only one component while leaving the other unchanged. Fig. 2(c) gives an overview of the resulting graph. If the factor graphs are d_1- and d_2-regular, with spectral gaps δ_1 and δ_2, then the spectral gap δ of the product graph satisfies $\delta \geq \min\{\delta_1 d_1,\ \delta_2 d_2\}/(d_1 + d_2)$ [16].

Gate-based QW realization. The first gate-based quantum circuit realizing the MNRS QW framework was introduced in [19] and later extended in [20]. To prepare the initial uniform superposition of edges required by Eq. 3, the construction starts by first generating a uniform superposition of vertices $\in \mathcal{V}$ through the $U_{\mathcal{V}}$ operator. The first step of $U_{\mathcal{V}}$ is the generation of a Dicke state $|D_k^n\rangle$, corresponding to the uniform superposition of the $\binom{n}{k}$ length-n bitstrings of Hamming weight k. This state is prepared using the deterministic circuit of [2].

The transition from a uniform superposition of bitstrings to a uniform superposition over indices $\in \mathcal{V}$ is enabled by the BIX operator proposed in [19], which was later refined and renamed the *Vertex Binary Encoder (VBE)* in [20]. Operationally, the VBE maps each bitstring encoded in the basis states of $|D_k^n\rangle$ to two ordered lists of indices: S, containing the positions of the k bits set to 1, and S^*, containing the remaining $n-k$ positions set to 0. These lists are stored in the quantum registers $\underline{S}$ and $\underline{S^*}$, respectively. As shown in Fig. 3(a), the VBE processes the input register sequentially, inserting the current counter value into the appropriate list whenever a 1 is detected, using cyclic shifts. Applying the same procedure to the negated input register produces the complementary list S^*.

Once the vertex has been encoded in index form, the gate-based realization of the update operator U_u acts directly on $\underline{S}$ and $\underline{S^*}$ to generate neighboring vertices. Two auxiliary arrays, $\underline{T}$ and $\underline{T^*}$, are initialized as copies of $\underline{S}$ and $\underline{S^*}$, respectively. Then, as illustrated in Fig. 3(b), a neighboring vertex is obtained by selecting one element from $\underline{T}$ and one from $\underline{T^*}$ uniformly at random and swapping them. The uniform selection is implemented using W-state generation circuits, denoted by $\mathtt{W_n}$, which correspond to uniform superpositions over length-n bitstrings of Hamming weight one, while insertion and deletion operations on the sorted index arrays are performed using the circuit proposed in [14].

3 Implementation

This section presents a detailed construction of the update operator, correcting errors in earlier proposals, and describes the three quantum strategies we employ to implement collision-based ISD within the QW search paradigm.

3.1 Quantum Walk Update Operator

Our QW construction follows the MNRS framework [23] and builds upon the gate-level subcircuits introduced in [19, 20, 28]. We refine these prior approaches by introducing a revised circuit-level realization of the update operator that eliminates conceptual inconsistencies and faithfully implements the intended QW dynamics.

In particular, earlier constructions implement the update operator in a manner that departs from its mathematical definition in [23]. Since the update operator is applied as part of each QW step, all ancillary registers (namely $\underline{\boldsymbol{\alpha}}$, $\underline{\boldsymbol{\alpha}^*}$, $\underline{\boldsymbol{\omega}}$, and $\underline{\boldsymbol{\omega}^*}$) must be restored to the $|0\rangle$ state before the next step is executed. In [20], this cleanup is attempted by directly applying $U_u^\dagger$. However, this approach is incorrect because the auxiliary registers $\underline{\boldsymbol{\omega}}$ and $\underline{\boldsymbol{\omega}^*}$, which encode the W-state used for sampling, remain entangled with the registers representing the source and destination vertices. As a result, the ancillae are not properly disentangled, and subsequent walk steps do not start from a clean state, causing the circuit evolution to deviate from the intended QW behavior. To address this issue, we introduce an explicit uncomputation step that restores all auxiliary registers without disturbing the walk state. The proposed procedure should be executed at the end of the update operator proposed in [20], and exemplified in Fig. 3(b). The reset circuit is shown in Fig. 4. It combines direct comparisons between the register $\underline{\boldsymbol{\alpha}}$ and the elements of $\underline{S}$ with a membership test for sorted arrays between the elements of $\underline{S}$ and $\underline{T}$, implemented using the circuit of [14].

Operationally, to correctly reset $\underline{\boldsymbol{\omega}}$ (respectively, $\underline{\boldsymbol{\omega}^*}$), we sequentially compare the content of $\underline{\boldsymbol{\alpha}}$ ($\underline{\boldsymbol{\alpha}^*}$) with each element of $\underline{T}$ ($\underline{T^*}$). A match identifies the index selected during the sampling step and therefore certifies that the corresponding qubit of $\underline{\boldsymbol{\omega}}$ ($\underline{\boldsymbol{\omega}^*}$) was in the state $|1\rangle$. This qubit can then be reset by applying a single NOT gate, after which the comparison is uncomputed to restore $\underline{\boldsymbol{\alpha}}$ ($\underline{\boldsymbol{\alpha}^*}$) and $\underline{T}$ ($\underline{T^*}$) to their original values.

Each comparison, depicted as $\stackrel{?}{=}$ in Fig. 4, is implemented by XORing each element of $\underline{T}$ ($\underline{T^*}$) with $\underline{\boldsymbol{\alpha}}$ ($\underline{\boldsymbol{\alpha}^*}$) through a sequence of controlled NOT gates. If the resulting value in $\underline{\boldsymbol{\alpha}}$ ($\underline{\boldsymbol{\alpha}^*}$) is the all-zero bitstring, the two registers match and a $|1\rangle$ is written to an auxiliary qubit through a multi-controlled NOT gate. This qubit is then used as a control to reset the corresponding qubit of $\underline{\boldsymbol{\omega}}$ ($\underline{\boldsymbol{\omega}^*}$) via a controlled NOT gate. The comparison is subsequently uncomputed to reset both $\underline{\boldsymbol{\alpha}}$ ($\underline{\boldsymbol{\alpha}^*}$) and the auxiliary register to their original state before the next comparison.

At the end of the sequence of comparisons, we need to restore the ancillary register $\underline{\boldsymbol{\alpha}}$ itself to the zero state. For this operation, we exploit the relationship between $\underline{S}$ and $\underline{T}$. At the beginning of the update step, these registers contain identical values, while at the end of the sampling procedure they differ by exactly one element. For each element $\underline{S}[i]$, we test whether it appears in $\underline{T}$ using the membership test of [14], storing the Boolean outcome in an ancilla qubit. If the outcome indicates that $\underline{S}[i]$ is not present in $\underline{T}$, then this element must coincide with the value stored in $\underline{\boldsymbol{\alpha}}$. The ancilla qubit is then used as a control to XOR $\underline{S}[i]$ into $\underline{\boldsymbol{\alpha}}$, restoring it to $|0\rangle$ while leaving $\underline{S}[i]$ unchanged. Finally, the ancilla qubit is uncomputed by reversing the membership test, which is safe since $\underline{S}$ and $\underline{T}$ remain unchanged during the whole procedure. An analogous procedure is applied to reset $\underline{\boldsymbol{\alpha}^*}$ using $\underline{S^*}$ and $\underline{T^*}$.

The resulting update operator correctly implements the transition dynamics of the Johnson graph while operating entirely on the index representations. Resource estimates for the full update operator, including gate counts, circuit depth, and qubit requirements, are reported in Tab. 1.

3.2 Quantum walk for ISD attacks

A quantum realization of Stern's strategy, introduced in Sec. 2.1, naturally decomposes into two conceptually distinct search tasks. The first task searches for a suitable information set $\mathcal{I}$, while the second task implements the collision-finding procedure that constitutes the core of Stern's algorithm. At a high level, the first search

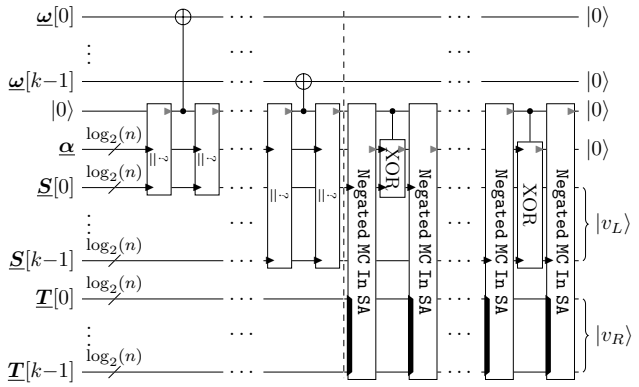


Figure 4: Circuit used to restore $|w\rangle$ and $|\alpha\rangle$ to $|0\rangle$. MC is the Membership Check circuit of [14]. Convetions as per Fig. 3(b).

Table 1: (a) Complexity measures for the update U_u of the Johnson graph $J(n,k)$, incorporating the uncomputation circuits introduced in Sec. 3.1, and refining the original construction of [20]. Here $m = \lceil \log_2(n) \rceil$, and $r = n-k$.

	Copy	W-state Sample	Del/Ins	Reset W-state	Clear $\underline{\alpha}$
X	0	2	$4nm$	$2nm$	$4n^2m$
R_Y	0	$n-1$	0	0	0
CNOT	nm	$n-1$	$22nm+$ $-18n+$ $+2m$	$2nm$	$8n^2m$ $+4n^2-2n$
CCNOT	0	nm	$8nm-4n$	0	n
C^mX	0		0	n	$4n^2$
CSWAP	0		$2nm-2m$	0	0
Depth	1	$nm-km+$ $+2\log_2(r)+$ $+2$	$2\log_2(r)+$ $+20m-2$	$5r$	$8n\log_2(r)+19n+$ $-8k\log_2(r)-19k$
Qubits	$2nm$	$2m$	$2nm+2n$	2	0

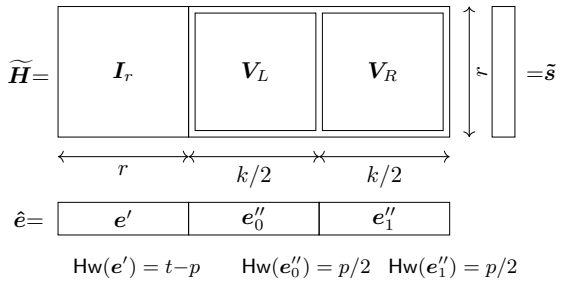


Figure 5: Modified Stern configuration, adapted for a collision search in the quantum setting.

replaces the QAA-based construction of [30] with a QW over candidate $\mathcal{I}$, while the second search can be realized in different ways. We propose three quantum strategies that realize this two-search

QS	Qubits	Quantum iterations	U_f depth
QS0	$O(r^2)$	$\sqrt{\frac{\binom{n}{t}}{\binom{r}{t-p}\binom{k/2}{p/2}^2}}$	$O(r^2)+\binom{k/2}{p/2}^2 O\left(\log_2^2(r)\right)$
QS1	$O(r^2)$	$\sqrt{\frac{\binom{n}{t}}{\binom{r}{t-p}\binom{k/2}{p/2}^2}\binom{k/2}{p/2}^2}$	$O(r^2)+O\left(\log_2^2(r)\right)$
QS2	$O(r^2)$	$\sqrt{\frac{\binom{n}{t}}{\binom{r}{t-p}\binom{k/2}{p/2}^2}\binom{k/2}{p/2}}$	$O(r^2)+\binom{k/2}{p/2}O\left((\log_2^2(r))\right)$

Table 2: Asymptotic costs of the three quantum strategies (QS) proposed to adapt the Stern's variant of ISD to a collision-finding in the quantum paradigm.

decomposition with different trade-offs between the complexity of the QW and the cost of the associated oracle. Tab. 2 gives the asymptotic complexity measures for all these strategies.

Adapting the classical algorithm to the quantum setting requires a substantial reinterpretation of several steps. In particular, reversible quantum computation forbids early aborts: branches of the superposition for which the QGJE does not produce an identity matrix on its leftmost block cannot be discarded, and all subsequent operations must be applied, independently from this condition being satisfied, uniformly to both successful and unsuccessful branches.

Similarly, the classical separation between collision-search filtering on the first l coordinates (lines 9–14), followed by a final Hamming-weight check on the remaining coordinates when a collision occurs (line 14), cannot be preserved, since all computations must proceed coherently on the entire superposition. For this reason, the traditional top–bottom partition of the matrix $\boldsymbol{V}$ and of the syndrome vector $\tilde{\boldsymbol{s}}$ provides no computational advantage in the quantum model. As a consequence, the corresponding split of the partial error vector $\boldsymbol{e}'$ becomes ineffective in the quantum setting. By contrast, the left–right partition of $\boldsymbol{V}$ and of $\boldsymbol{e}''$ remains an essential tool to enable a collision search between the two halves of the error vector. This modified structure is illustrated in Fig. 5.

Finally, we stress that multiplying $\boldsymbol{V}_L$ (respectively, $\boldsymbol{V}_R$) by the binary error vector $\boldsymbol{e}''_0$ (respectively, $\boldsymbol{e}''_1$) is equivalent to taking the bitwise XOR of the columns of the matrix whose indices correspond to the positions where the vector equals 1. This correspondence, already apparent in the classical algorithm (see line 10 and 13), carries over directly to the quantum setting: it is implemented as a controlled selection and summation of the quantum registers encoding the columns of $\boldsymbol{V}_L$ and $\boldsymbol{V}_R$, with the controls given by the qubits of the candidate error vector $\boldsymbol{e}''$ that are in the $|1\rangle$ state.

Quantum strategy 0 (QS0). The first strategy consists of a single QW over the superposition of all possible choices of the information set $\mathcal{I}$, combined with a modified oracle that, after the QGJE, explicitly enumerates all candidate partial error patterns within the walk oracle. For this latter enumeration, a first quantum registers is used to encode the generation, one at a time, of all $\binom{k/2}{p/2}$ bitstrings, each one acting as one-hot encoding of the candidate set of indices $\mathcal{J}_0$. For each bitstring, a second register contains the enumeration of the additional $\binom{k/2}{p/2}$ bitstrings, each one providing the one-hot encoding of a candidate set of indices $\mathcal{J}_1$. For each pair of bitstrings

$(j_0, j_1) \in \mathcal{J}_0 \times \mathcal{J}_1$, their asserted qubits act as controls to sum the corresponding columns of the matrix $\boldsymbol{V}$ into the register holding the syndrome $\tilde{\boldsymbol{s}}$.

To optimize the depth of the circuit, the enumeration strategy follows the approach introduced in [30]. Since the value of p is known at quantum circuit generation time, each candidate set $\mathcal{J}_0$ (and $\mathcal{J}_1$) is precomputed by the classical controller, that generates in turn a sequence of appropriate quantum gates. Additionally, to optimize the complexity metrics of the circuit, each candidate set differ from the following one by only a single index, corresponding to a single column of $\boldsymbol{V}_L$ ($\boldsymbol{V}_R$), exploiting a *revolving door* enumeration of combinations of k elements out of n [18, Chap. 7.2.1.3], which allows the partial sums to be updated incrementally rather than recomputed from scratch.

As a consequence, the oracle executes sequentially the Hamming-weight checks for all $\binom{k/2}{p/2}^2$ possible pairs of partial sums, each sum realized through a quantum circuit having depth $O\left(\log_2^2(r)\right)$ [29]. The cost of a single iteration is therefore balanced between the cost of executing the QGJE, whose depth scales as r^2 [31], and the pairwise summation and checking of the $\binom{k/2}{p/2}^2$ candidate vectors, having cost $\approx \binom{k/2}{p/2}^2 \log_2^2(r)$. The number of QW iterations is given by the square root of the inverse success probability from Eq. 2, taking into account the fact that there is no restrictions on the length-l run of zeros on the first l coordinates of $\boldsymbol{e}'$.

Quantum strategy 1 (QS1). At the opposite end of the design spectrum, this strategy implements the full Stern-style search as a single QW over an expanded combinatorial space. Instead of relegating the collision search to the oracle, QS1 embeds it directly into the walk dynamics by operating on the Cartesian product of three Johnson graphs: $J(n,k) \otimes J_0\left(\frac{k}{2}, \frac{p}{2}\right) \otimes J_1\left(\frac{k}{2}, \frac{p}{2}\right)$. The graph $J(n,k)$ encodes a uniform superposition of all the candidate information sets $\mathcal{I}$ and governs the QGJE performed coherently on the register encoding the parity-check matrix $\boldsymbol{H}$. The remaining graphs J_0 and J_1 span the $\binom{k/2}{p/2}$ binary vectors of weight $p/2$ and length $k/2$, thereby representing all possible partial error patterns on the two halves of $\boldsymbol{e}''$. Their joint evolution realizes a coherent meet-in-the-middle search, which detects collisions between the two partial sums on the quantum registers encoding $\boldsymbol{V}_L$ and $\boldsymbol{V}_R$; when a collision occurs, the circuit verifies the associated Hamming-weight constraint on the remaining portion of the error vector.

During input preparation, the quantum registers are initialized in a uniform superposition over the vertices of the three Johnson graphs. The QW then evolves coherently over all triplets $(j, j_0, j_1) \in J \times J_0 \times J_1$. At the oracle stage, the indices specified by J_0 and J_1 are used to select and sum the corresponding columns of $\boldsymbol{V}_L$ and $\boldsymbol{V}_R$ into auxiliary registers. The oracle applies a global phase flip if and only if the resulting partial sums collide and the combined vector has the target Hamming weight.

With this construction, the cost of a single iteration is dominated by the QGJE, whose circuit depth scales as r^2 [31]; the overhead of forming partial sums is asymptotically negligible. By contrast, the walk space is enlarged by a factor $\binom{k/2}{p/2}^2$, due to the two additional Johnson graphs, which increases the number of QW iterations by the square root of this factor.

Relative to QS0, QS1 transfers computational effort from the oracle to the walk itself: the oracle is simpler, but the walk operates over a substantially larger state space, resulting in a higher overall walk complexity.

Quantum strategy 2 (QS2). The third strategy explores an intermediate design point between QS0 and QS1. Rather than preparing a superposition over both halves of the error vector, QS2 performs the collision search coherently over a single Johnson graph, while the second half is processed exhaustively within the oracle as per QS0. As in the previous strategies, the search for a valid information set is implemented via a quantum walk.

Accordingly, the walk space is given by the Cartesian product of two Johnson graphs, $J(n,k) \otimes J_0\left(\frac{k}{2}, \frac{p}{2}\right)$, where $J(n,k)$ indexes all candidate information sets $\mathcal{I}$ and drives the QGJE applied in superposition to the register encoding the matrix $\boldsymbol{H}$. The graph J_0 spans the $\binom{k/2}{p/2}$ error patterns of weight $p/2$ supported on $\boldsymbol{V}_L$, and is used to perform one side of the collision search.

During input preparation, the quantum registers are initialized in a uniform superposition over the vertices of the two Johnson graphs, and the walk explores all pairs $(j, j_0) \in J \times J_0$ coherently. Inside the oracle, after executing the QGJE, the indices specified by J_0 are used to select and XOR the corresponding columns of $\boldsymbol{V}_L$ into the register holding the syndrome $\tilde{\boldsymbol{s}}$. The remaining half of the collision search is carried out by exhaustively enumerating all $\binom{k/2}{p/2}$ choices of columns of $\boldsymbol{V}_R$, following the revolving-door enumeration strategy already employed in QS0. This construction more closely mirrors the classical meet-in-the-middle structure of Stern's algorithm: the classical associative data structure is replaced by a uniform superposition over partial sums, while the second half of the search is handled deterministically within the oracle.

As in the previous strategies, the cost of a single iteration is dominated by the QGJE. Compared to QS1, the walk space is reduced by a factor $\binom{k/2}{p/2}$, which decreases the number of QW iterations by the square root of this factor. In return, the oracle evaluates the Hamming-weight and checking circuits only $\binom{k/2}{p/2}$ times per iteration, yielding a different balance between oracle depth and walk complexity.

4 Experimental Results

To assess the computational cost of our approach under cryptographically meaningful parameters, we consider the code-based cryptographic schemes advanced to the fourth round of NIST PQC standardization process [1]: Classic McEliece, BIKE, and HQC. In line with the submission requirements [26], these schemes are defined at three target security levels—L1, L3, and L5—corresponding to a quantum attack cost comparable to exhaustive key search on AES-128, AES-192, and AES-256, respectively. The concrete parameter are summarized in Table 3.

NIST's evaluation methodology adopts a uniform cost model in which all quantum gates are assigned equal weight. More recent work, however, advocates a finer-grained analysis in which gates acting on more than two qubits are decomposed into circuits composed solely of one- and two-qubit gates, with the `CCNOT` as the only allowed three-qubit primitive. Following this approach, we translate all multi-controlled operations in our construction

Table 3: Complexity estimates for the quantum walk (QW) implementation of Stern's strategy applied to BIKE, HQC, and Classic McEliece, compared with the quantum ISD attack of [19]. Except for the code parameters and p, all values are in $\log_2$.

Scheme	Level	Code parameters			QW	This work						[30]					
		n	k	t	iterations	p	G	D	W	D×W	G×D	p	G	D	W	D×W	G×D
BIKE (key)	L1	24 646	12 323	142	65	2	109	93	29	122	203	1	105	90	29	119	195
	L3	49 318	24 659	206	96	2	144	127	31	158	271	1	140	123	31	154	263
	L5	81 946	40 973	274	130	2	180	162	33	194	342	1	175	158	33	191	333
BIKE (message)	L1	24 646	12 323	134	61	2	105	89	29	118	195	1	102	86	29	115	188
	L3	49 318	24 659	199	93	2	140	123	31	154	264	1	136	120	31	151	256
	L5	81 946	40 973	264	125	2	175	157	33	189	332	1	170	153	33	186	323
HQC	L1	35 338	17 669	132	60	2	106	89	30	119	195	1	102	86	30	116	188
	L3	71 702	35 851	200	93	2	142	125	32	157	267	1	138	121	32	153	259
	L5	115 274	57 637	262	124	2	175	157	34	190	332	1	171	153	34	186	324
Classic McEliece	L1	3 488	2 720	64	70	2	100	91	22	112	191	1	97	89	22	110	186
	L3	4 608	3 360	96	91	2	123	113	23	136	236	1	120	111	23	134	231
	L5	6 688	5 024	128	129	2	162	152	24	176	315	1	160	150	24	174	310
	L5	6 960	5 413	119	124	2	162	155	24	179	317	1	160	150	24	174	310
	L5	8 192	6 528	128	142	2	181	174	24	198	354	1	179	169	24	193	348

into logarithmic-depth circuits using the method of [29], which yields the most favorable performance in our setting. Although this decomposition introduces a number of ancillary qubits linear in the number of controls, all such ancillas are uncomputed at the end of each operation, and the overall qubit count remains unchanged.

Tab. 3 reports the estimated quantum resources required to mount attacks against the three schemes under consideration, when using QS2, which was experimentally validated as the best one for this range of parameters. Assuming a uniform gate-cost model, we report the total number of gates (G), circuit depth (D), and circuit width (W). We additionally include the depth–width product (D×W), introduced in [14], which captures common trade-offs between parallelism and circuit depth. Finally, we report the gate–depth product (G×D), corresponding to the QAES metric used by NIST to estimate the cost of quantum attacks when circuit depth is bounded and parallel execution is required. The table also provides a direct comparison with quantum attack to the SDP proposed in [30], which is based on the Lee–Brickell variant [21]. This approach improve upon earlier Prange-based constructions [8, 29], all of which rely on the QAA framework. Our experimental evaluation further indicates that the optimal choice of the parameter p, across all considered metrics, is $p = 2$, corresponding to a balanced decomposition of the error vector into two components of weight 1. This observation is consistent with the optimal value reported in [19].

Overall, the results in Tab. 3 show that, unlike in the classical setting, the quantum-walk adaptation of the collision-based strategy does not yield improved asymptotic complexity. In particular, the circuit depth increases by a factor between 2^3 and 2^4, with a comparable increase in the total gate count. The width, on the other hand, remains in the same range.

5 Conclusion

We proposed a revised realization of the update operator for the quantum walk framework, correcting issues that were overlooked in previous constructions. Building on this, we investigated several techniques for lifting collision-based ISD attacks into the quantum setting, including the use of quantum walks over Cartesian products of Johnson graphs to accelerate the search. Our analysis shows that QW-based approaches outperform plain QAA only for specific classes of problems, a behavior that can be explained by the relative costs of the underlying algorithmic components. Indeed, for structured search problems, a quantum walk can provide an advantage over QAA only when the cost of the oracle is sufficiently small compared to the cost of the walk itself. Existing results [23] establish that quantum walks improve upon QAA when the cost of input preparation exceeds $1/\sqrt{\delta}$ times the cost of the update operator $\boldsymbol{U}_u$, where δ denotes the spectral gap of the underlying Markov chain. However, these analyses typically abstract away the cost of the oracle. Our results indicate that, for a quantum walk to be balanced in practice, the oracle cost must be comparable to that of the update operator. Equivalently, the complexity of $\boldsymbol{U}_f$ should lie in the same asymptotic regime as that of $\boldsymbol{U}_u$. Under our circuit-level realization of the update operator, this condition translates into the oracle having depth at most $n \log n$, where n is the size of the Johnson graph $J(n, k)$. Only in this regime the quantum-walk approach yields a concrete advantage over a plain QAA-based strategy. We leave for future work the adaptation of our quantum-walk circuits to other relevant problems, such as the k-sum problem, as well as the exploration of hybrid strategies for collision-based ISD approaches.

Acknowledgments

This work was partially supported by the collaborative project *SPQR (Smart Precision for Quantum Resilience)*, which is funded by the Norwegian Research Council (RCN) through the TEKNOLOGIKONVER program. Project number: 358698.

References

[1] Gorjan Alagic, D Apon, D Cooper, Q Dang, T Dang, J Kelsey, J Lichtinger, C Miller, D Moody, R Peralta, et al. [n. d.]. Status Report on the Fourth Round of the NIST Post-Quantum Cryptography Standardization Process. , 34 pages. https://doi.org/10.6028/NIST.IR.852

[2] Andreas Bärtschi and Stephan J. Eidenbenz. [n. d.]. Deterministic Preparation of Dicke States. In *Fundamentals of Computation Theory - 22nd International Symposium, FCT 2019, Copenhagen, Denmark, August 12-14, 2019, Proceedings* (2019) *(Lecture Notes in Computer Science, Vol. 11651)*, Leszek Antoni Gasieniec, Jesper Jansson, and Christos Levcopoulos (Eds.). Springer, 126–139. https://doi.org/10.1007/978-3-030-25027-0_9

[3] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. [n. d.]. Decoding Random Binary Linear Codes in $2^{n/20}$: How 1 + 1 = 0 Improves Information Set Decoding. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings* (2012) *(Lecture Notes in Computer Science, Vol. 7237)*, David Pointcheval and Thomas Johansson (Eds.). Springer, 520–536. https://doi.org/10.1007/978-3-642-29011-4_31

[4] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. [n. d.]. On the Inherent Intractability of Certain Coding Problems (Corresp.). 24, 3 ([n. d.]), 384–386. https://doi.org/10.1109/TIT.1978.1055873

[5] Daniel J. Bernstein. [n. d.]. Grover vs. McEliece. In *Post-Quantum Cryptography*, Nicolas Sendrier (Ed.). Vol. 6061. Springer Berlin Heidelberg, 73–80. https://doi.org/10.1007/978-3-642-12929-2_6

[6] Leif Both and Alexander May. [n. d.]. Decoding Linear Codes with High Error Rate and Its Impact for LPN Security. In *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings* (2018) *(Lecture Notes in Computer Science, Vol. 10786)*, Tanja Lange and Rainer Steinwandt (Eds.). Springer, 25–46. https://doi.org/10.1007/978-3-319-79063-3_2

[7] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. [n. d.]. Quantum Amplitude Amplification and Estimation. ([n. d.]), 53–74. https://doi.org/10.1090/conm/305/05215

[8] Clémence Chevignard, Pierre-Alain Fouque, and André Schrottenloher. [n. d.]. Reducing the Number of Qubits in Quantum Information Set Decoding. In *Advances in Cryptology – ASIACRYPT 2024: 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9–13, 2024, Proceedings. Part VIII* (Kolkata, India and Berlin, Heidelberg, 2024). Springer-Verlag, 299–329. https://doi.org/10.1007/978-981-96-0944-4_10

[9] Ilya Dumer. [n. d.]. On Minimum Distance Decoding of Linear Codes. In *Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory* (1991). Moscow, 50–52.

[10] Matthieu Finiasz and Nicolas Sendrier. [n. d.]. Security Bounds for the Design of Code-Based Cryptosystems. In *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings* (2009) *(Lecture Notes in Computer Science, Vol. 5912)*, Mitsuru Matsui (Ed.). Springer, 88–105. https://doi.org/10.1007/978-3-642-10366-7_6

[11] Lov K. Grover. [n. d.]. A Fast Quantum Mechanical Algorithm for Database Search. In *Proceedings of the Twenty-Eighth Annual {ACM} Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996* (1996), Gary L. Miller (Ed.). ACM, 212–219. https://doi.org/10.1145/237814.237866

[12] Qian Guo, Thomas Johansson, and Vu Nguyen. [n. d.]. A New Sieving-Style Information-Set Decoding Algorithm. 70, 11 ([n. d.]), 8303–8319. https://doi.org/10.1109/TIT.2024.3457150

[13] Samuel Jaques and Arthur G. Rattew. [n. d.]. QRAM: A Survey and Critique. 9 ([n. d.]), 1922. https://doi.org/10.22331/q-2025-12-02-1922

[14] Samuel Jaques and John M. Schanck. [n. d.]. Quantum Cryptanalysis in the RAM Model: Claw-finding Attacks on SIKE. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I* (2019) *(Lecture Notes in Computer Science, Vol. 11692)*, Alexandra Boldyreva and Daniele Micciancio (Eds.). Springer, 32–61. https://doi.org/10.1007/978-3-030-26948-7_2

[15] Ghazal Kachigar and Jean-Pierre Tillich. [n. d.]. Quantum Information Set Decoding Algorithms. In *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings* (2017) *(Lecture Notes in Computer Science, Vol. 10346)*, Tanja Lange and Tsuyoshi Takagi (Eds.). Springer, 69–89. https://doi.org/10.1007/978-3-319-59879-6_5

[16] Ghazal Kachigar and Jean-Pierre Tillich. [n. d.]. Quantum Information Set Decoding Algorithms. arXiv:1703.00263 [cs.CR] https://arxiv.org/abs/1703.00263

[17] Elena Kirshanova. [n. d.]. Improved Quantum Information Set Decoding. In *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings* (2018) *(Lecture Notes in Computer Science, Vol. 10786)*, Tanja Lange and Rainer Steinwandt (Eds.). Springer, 507–527. https://doi.org/10.1007/978-3-319-79063-3_24

[18] Donald E Knuth. [n. d.]. *The Art of Computer Programming, Volume IVa: Combinatorial Algorithms.* Addison-Wesley Professional.

[19] Giacomo Lancellotti, Simone Perriello, Alessandro Barenghi, and Gerardo Pelosi. [n. d.]. Design of a Quantum Walk Circuit to Solve the Subset-Sum Problem. In *Proceedings of the 61st ACM/IEEE Design Automation Conference, DAC 2024, San Francisco, CA, USA, June 23-27, 2024* (2024), Vivek De (Ed.). ACM, 298:1–298:6. https://doi.org/10.1145/3649329.3657337

[20] Giacomo Lancellotti, Simone Perriello, Alessandro Barenghi, and Gerardo Pelosi. [n. d.]. Solving the Subset Sum Problem via Quantum Walk Search. ([n. d.]). https://doi.org/10.1109/TC.2025.3625044

[21] P. J. Lee and E. F. Brickell. [n. d.]. An Observation on the Security of McEliece's Public-Key Cryptosystem. In *Advances in Cryptology — EUROCRYPT '88*, D. Barstow, W. Brauer, P. Brinch Hansen, D. Gries, D. Luckham, C. Moler, A. Pnueli, G. Seegmüller, J. Stoer, N. Wirth, and Christoph G. Günther (Eds.). Vol. 330. Springer Berlin Heidelberg, 275–280. https://doi.org/10.1007/3-540-45961-8_25

[22] Jeffrey S. Leon. [n. d.]. A Probabilistic Algorithm for Computing Minimum Weights of Large Error-Correcting Codes. 34, 5 ([n. d.]), 1354–1359. https://doi.org/10.1109/18.21270

[23] Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. [n. d.]. Search via Quantum Walk. 40, 1 ([n. d.]), 142–164. https://doi.org/10.1137/090745854

[24] Alexander May, Alexander Meurer, and Enrico Thomae. [n. d.]. Decoding random linear codes in $\tilde{O}(20.054n)$. In *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings* (2011) *(Lecture Notes in Computer Science, Vol. 7073)*, Dong Hoon Lee and Xiaoyun Wang (Eds.). Springer, 107–124. https://doi.org/10.1007/978-3-642-25385-0_6

[25] Alexander May and Ilya Ozerov. [n. d.]. On Computing Nearest Neighbors with Applications to Decoding of Binary Linear Codes. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I* (Berlin, Heidelberg, 2015) *(Lecture Notes in Computer Science, Vol. 9056)*, Elisabeth Oswald and Marc Fischlin (Eds.). Springer, 203–228. https://doi.org/10.1007/978-3-662-46800-5_9

[26] National Institute of Standards and Technology (NIST). [n. d.]. Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process. https://csrc.nist.gov/pqc-standardization

[27] National Institute of Standards and Technology (NIST). [n. d.]. Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf

[28] Simone Perriello. [n. d.]. Quantum Circuit Design for Finding K-Cliques via Quantum Amplitude Amplification Strategies. In *Proceedings of the 22nd ACM International Conference on Computing Frontiers, CF 2025, Cagliari, Italy* (2025). https://doi.org/10.1145/3719276.3725200

[29] Simone Perriello, Alessandro Barenghi, and Gerardo Pelosi. [n. d.]. Improving the Efficiency of Quantum Circuits for Information Set Decoding. 4, 4 ([n. d.]), 1–40. https://doi.org/10.1145/3607256

[30] Simone Perriello, Alessandro Barenghi, and Gerardo Pelosi. [n. d.]. Quantum Circuit Design for the Lee-Brickell Based Information Set Decoding. In *Applied Cryptography and Network Security Workshops - ACNS 2024 Satellite Workshops, Aiblock, AIHWS, AIoTS, SCI, AAC, SiMLA, LLE, and CIMSS, Abu Dhabi, United Arab Emirates, March 5-8, 2024, Proceedings, Part II* (2024) *(Lecture Notes in Computer Science, Vol. 14587)*, Martin Andreoni (Ed.). Springer, 8–28. https://doi.org/10.1007/978-3-031-61489-7_2

[31] Simone Perriello, Alessandro Barenghi, and Gerardo Pelosi. [n. d.]. A Quantum Circuit to Speed-up the Cryptanalysis of Code-Based Cryptosystems. In *Security and Privacy in Communication Networks - 17th EAI International Conference, SecureComm 2021, Virtual Event, September 6-9, 2021, Proceedings, Part II* (2021) *(Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 399)*, Joaquín García-Alfaro, Shujun Li, Radha Poovendran, Hervé Debar, and Moti Yung (Eds.). Springer, 458–474. https://doi.org/10.1007/978-3-030-90022-9_25

[32] E. Prange. [n. d.]. The Use of Information Sets in Decoding Cyclic Codes. 8, 5 ([n. d.]), 5–9. https://doi.org/10.1109/TIT.1962.1057777

[33] Miklos Santha. [n. d.]. Quantum Walk Based Search Algorithms. In *Theory and Applications of Models of Computation, 5th International Conference, TAMC 2008, Xi'an, China, April 25-29, 2008. Proceedings* (2008) *(Lecture Notes in Computer Science, Vol. 4978)*, Manindra Agrawal, Ding-Zhu Du, Zhenhua Duan, and Angsheng Li (Eds.). Springer, 31–46. https://doi.org/10.1007/978-3-540-79228-4_3

[34] Peter W. Shor. [n. d.]. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994* (1994). IEEE Computer Society, 124–134. https://doi.org/10.1109/SFCS.1994.365700

[35] Jacques Stern. [n. d.]. A Method for Finding Codewords of Small Weight. In *Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings* (1988) *(Lecture Notes in Computer Science, Vol. 388)*, Gérard D. Cohen and Jacques Wolfmann (Eds.). Springer, 106–113. https://doi.org/10.1007/BFb0019850