Quantum Walks Design of Collision-Based Information Set Decoding

Simone Perriello[1], Alessandro Barenghi[1], and Gerardo Pelosi[1]

[1] *Politecnico di Milano — Department of Electronics, Information and Bioengineering*

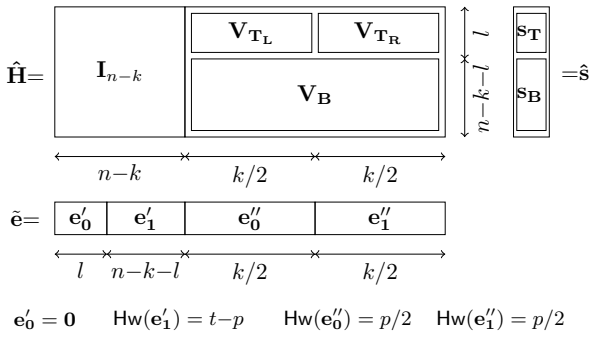

Figure 1: Weight distribution assumed in Stern algorithm.

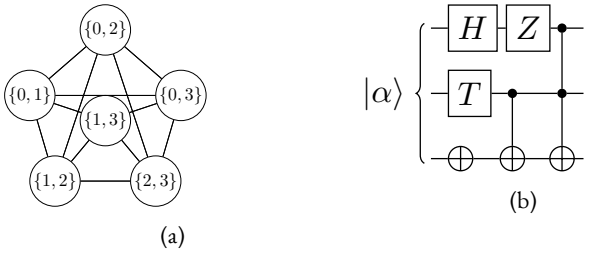

Figure 2: (a) Johnson graph $J(4,2)$. (b) Quantum circuit example.

Introduction. Code-based cryptography plays a central role in post-quantum cryptography, with its security relying on the hardness of the Syndrome Decoding Problem. The most effective attacks are variants of Information Set Decoding (ISD), which remain exponential-time even in the quantum setting. In this work, we propose a hybrid classical–quantum strategy for Stern-like collision-based ISD, where the quantum component exploits the quantum walk framework to perform the collision search. We evaluate our approach on cryptosystems that reached the final stages of the National Institute of Standards and Technology (NIST) standardization process, and show that, under conservative assumptions, it can outperform classical strategies when realistic classical memory-access costs are taken into account.

Collision based Information Set Decoding. We consider a binary linear code $\mathcal{C}$ of length n and dimension k, defined by a parity-check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$ such that $H\mathbf{c} = \mathbf{0} \in \mathbb{F}_2^{n-k}$ for every vector $\mathbf{c} \in \mathcal{C}$, called a *codeword*. Given a vector $\mathbf{y} = \mathbf{c} \oplus \mathbf{e} \in \mathbb{F}_2^n$, where $\mathbf{e} \in \mathbb{F}_2^n$ is an *error vector*, the associated *syndrome* is defined as

$$\mathbf{s} = \mathbf{H}\mathbf{y} = \mathbf{H}\mathbf{e} \in \mathbb{F}_2^{n-k}. \tag{1}$$

The *Syndrome Decoding Problem (SDP)* asks, given random-looking H and $\mathbf{s}$, whether there exists a vector $\mathbf{e}$ of Hamming weight at most t s.t. $\mathbf{He}$=$\mathbf{s}$. Its decision version was proven NP-complete [3], and the search version is NP-hard via standard reductions. This hardness forms the basis of many code-based cryptographic constructions.

Information Set Decoding (ISD) algorithms solve SDP through a probabilistic Las-Vegas strategy. Each iteration selects a random set $\mathcal{I} \subset \{1,\dots,n\}$, $|\mathcal{I}|=k$, expected to index a negligible portion of the non-zero elements of $\mathbf{e}$. The columns of $\mathbf{H}$ indexed by $\mathcal{I}$ are permuted to the right using a permutation matrix $\mathbf{P}$, and Eq. (1) becomes $\mathbf{s} = (\mathbf{HP})(\mathbf{P}^{-1}\mathbf{e})$, which corresponds to solving the same SDP instance for a code that is permutation-equivalent to the original one. The augmented matrix $[(\mathbf{HP}) \mid \mathbf{s}]$ is transformed via *Gauss–Jordan Elimination (GJE)* into $\left[\hat{\mathbf{H}} \mid \hat{\mathbf{s}}\right]$, with $\hat{\mathbf{H}} = \left[\mathbf{I_r} \mid \mathbf{V}\right]$, with $\mathbf{V} \in \mathbb{F}_2^{(n-k)\times k}$. After this preprocessing step, ISD variants attempt to reconstruct $\mathbf{e}$ using different strategies.

Improving on Prange's original proposal [14], early approaches, such as Lee–Brickell [11], aim to increase the success probability of each iteration at the expense of a slight increase of the cost per iteration. More advanced algorithms instead, while reducing the success probability of a single iteration, aim to reduce the computational cost of a single iteration through time–memory trade-offs. In this respect, Stern's algorithm [15] is the first to employ exponential memory, introducing the *collision-based ISD* paradigm by turning the search for the error vector into a collision search on partial syndromes.

Fig. 1 illustrates the configuration assumed by Stern's ISD. Assuming that $p \ll t$ errors lie in $\mathcal{I}$, the corresponding portion of the error vector, denoted $\mathbf{e}''$, is split into two halves of size $k/2$, each assumed to have weight $p/2$. The remaining part, denoted $\mathbf{e}'$, is assumed to contain a run of l leading zeros, followed by $t-p$ ones in the remaining $n-k-l$ positions. Accordingly, $\hat{\mathbf{H}}$ is partitioned so that the right block $\mathbf{V}$ splits into a top part of l rows, V_T, and a bottom part of $n-k-l$ rows, $\mathbf{V_B}$. The top block is further divided into two halves, $\mathbf{V_{T_L}}$ and $\mathbf{V_{T_R}}$. Under these assumptions, the first l coordinates of the syndrome equation reduce to

$$\mathbf{s}_T = \bigoplus_{j\in\mathcal{J}_0} \mathbf{V_{T_L}}\big|_j \oplus \bigoplus_{j\in\mathcal{J}_1} \mathbf{V_{T_R}}\big|_j \implies \mathbf{s}_T \oplus \bigoplus_{j\in\mathcal{J}_0} \mathbf{V_{T_L}}\big|_j = \bigoplus_{j\in\mathcal{J}_1} \mathbf{V_{T_R}}\big|_j, \tag{2}$$

where $\mathcal{J}_0$ and $\mathcal{J}_1$ are size-$k/2$ sets used to index the selected columns in the two halves of $\mathbf{V_T}$. By exploiting a meet-in-the-middle strategy, Stern's algorithm enumerates all weight-$p/2$ patterns in the first half and stores the resulting partial syndromes in a list of size $\binom{k/2}{p/2}$. It then enumerates the patterns in the second half and searches for collisions with the stored values. Whenever a collision is found, the candidate is verified by checking whether $\mathbf{e}' = \mathbf{s_T} \oplus \left(\bigoplus_{j\in\mathcal{J}_0} \mathbf{V_{T_L}}\big|_j\right) \oplus \left(\bigoplus_{j\in\mathcal{J}_1} \mathbf{V_{T_R}}\big|_j,\right)$ has weight $t-p$.

Quantum Computing. Quantum algorithms operate on *qubits*. Using Dirac notation, a qubit state is written as $|\psi\rangle = a_0\,|0\rangle + a_1\,|1\rangle$, where $a_0, a_1 \in \mathbb{C}$, $|a_0|^2 + |a_1|^2 = 1$, and $|0\rangle, |1\rangle$ are basis vectors spanning the two-dimensional complex Hilbert space $\mathbb{C}^2$. An n-qubit system is described by a vector in a 2^n-dimensional Hilbert space, $|\psi\rangle = \sum_{i\in\{0,1\}^n} a_i\,|i\rangle$, with $|i\rangle$ denoting a basis state obtained as the tensor product of single-qubit states. Measuring the system returns the basis state $|i\rangle$ with probability $|a_i|^2$. The evolution of an n-qubit system is governed by *unitary operators* $\mathbf{U} \in \mathbb{C}^{2^n\times 2^n}$ satisfying $\mathbf{U}\mathbf{U}^\dagger = \mathbf{I}$. Applying $\mathbf{U}$ to $|\psi\rangle$ produces the new state $\mathbf{U}\,|\psi\rangle$.

Quantum algorithms are commonly described in the *circuit model*, where unitary transformations are decomposed into sequences of elementary *quantum gates*, similarly to classical Boolean circuits (see Fig. 2b). For example, the `NOT` gate (depicted as $\oplus$) flips a qubit between $|0\rangle$ and $|1\rangle$. The controlled-`NOT` flips a target qubit (shown as $\oplus$) when a control qubit (filled circle) is in state $|1\rangle$; the `CCNOT` gate adds a second control. The `Z` gate applies a phase flip to the state $|1\rangle$, mapping $|1\rangle$ to $-|1\rangle$ while leaving $|0\rangle$ unchanged. The `S` and `T` gates apply smaller phase shifts, mapping $|1\rangle$ to $e^{i\pi/2}\,|1\rangle$ and $e^{i\pi/4}\,|1\rangle$ respectively. Finally, the `H` gate creates the superposition $2^{-1/2}(|0\rangle + |1\rangle)$.

Quantum Walk. Given a domain $\mathcal{V}$ and a Boolean predicate $f : \mathcal{V} \to \{0,1\}$, with the goal of finding a marked element

$v \in \mathcal{M} = \{v \in \mathcal{V} \mid f(v) = 1\}$, a classical random walk explores a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ by moving between adjacent vertices according to a probabilistic distribution. A quantum walk extends this idea by evolving a superposition of edges of the graph using a unitary *walk operator* $\mathbf{U_w}$. The quantum walk algorithm, as given in the so-called MNRS model framework [12], starts by preparing the initial superposition of all graph edges through the setup operator $\mathcal{U}_s$, and then alternates two operators: 1) an oracle $\mathcal{U}_o$ that flips the phase of marked states, and 2) an approximate Householder reflection around the initial superposition, implemented via Quantum Phase Estimation on $\mathbf{U_w}$, the latter providing a unitary implementation of the classical transition matrix $\mathcal{P}$ associated to $\mathcal{G}$. A marked element is found after $\mathcal{O}(\sqrt{|\mathcal{V}|/|\mathcal{M}|})$ iterations, and each iteration uses one application of U_o, and $\mathcal{O}(1/\sqrt{\delta})$ applications of $\mathbf{U_w}$, with δ being the spectral gap of the graph. To model the domain space, quantum walks often use Johnson graphs $\mathcal{J}(n, k)$, whose vertices are k-subsets of $\{1, \ldots, n\}$, with edges connecting subsets differing by one element, as shown in Fig. 2b. They are $k(n-k)$ regular, and their spectral gap is $\delta = n/(k(n-k))$. Products of Johnson graphs naturally model joint combinatorial searches. 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 $\delta \geq \min\{\delta_1 d_1,\ \delta_2 d_2\}/\ (d_1 + d_2)$ [8].

Quantum walk — circuit implementation. Practical cost estimates of quantum walk search require explicit implementations of all the involved operators in terms of quantum gates. When using Johnson graphs, the setup operator $\mathbf{U_s}$ can be implemented through a Dicke-state generation circuit [2], as shown in [9, 10]. The oracle operator depends on the specific problem under consideration. The walk operator $\mathbf{U_w}$, on the other hand, relies on four applications of the update operator $\mathbf{U_u}$. While the state-of-the-art proposals in [9, 10] implement this operator using an integer representation of the Johnson graph vertices, we design two novel operators acting directly on the k-hot encoding of the graph. These design choices enable different trade-offs between circuit depth and width, and outperform [10] across all considered complexity metrics.

Quantum Walk Collision Search for ISD. Our hybrid approach to solving SDP with quantum adaptions of the ISD performs the permutation and GJE stages of ISD classically, and then delegates the search for the collision expressed in Eq. (2) to a quantum device. Specifically, whereas the classical algorithm performs a meet-in-the-middle step by storing the left side of the equation in a table for all possible sets $\mathcal{J}_0$ and computing the right side on the fly, our approach replaces this step with a quantum walk over the product $\mathcal{J}_0\left(\frac{k}{2}, \frac{p}{2}\right) \otimes \mathcal{J}_1\left(\frac{k}{2}, \frac{p}{2}\right)$, where each graph represents the selection of $p/2$ columns from one half of the matrix V. In this setting, the oracle operator evaluates the two sides of Eq. (2) independently and flips the phase whenever a collision is detected and $\mathtt{Hw}(\mathbf{e}') = t-p$. In this configuration, the walk requires $\binom{k/2}{p/2}$ iterations.

Complexity results. Tab. 1 reports the complexity metrics for the BIKE, HQC, and Classic McEliece cryptosystems, which reached the fourth round of the NIST competition [1]. The table includes the best known classical ISD attack, expressed in terms of the number of classical operations (*Op.s*) and memory size (M), assuming both a logarithmic memory access cost ($\log_2(M)$) or a square-root access cost ($\sqrt{M}$), the latter generally considered more realistic. These complexity estimates are obtained using the tool [7], by setting an optimistic upper bound on the classical memory size of 2^{80} bits. The table also reports the complexity metrics for the hybrid attack based on Stern's ISD algorithm. The D_{sc} column shows the maximum depth of a single quantum circuit. The remaining columns report aggregate metrics obtained by multiplying the metrics of a single quantum circuit and the expected number of times such a circuit must be run, which is given by $\binom{n}{t}$ (the number of vectors of length n and weight t), divided by $\binom{n-k}{t-p}\binom{k/2}{p/2}^2$ (which follows from the error distribution assumed by Stern's algorithm). All quantum metrics are evaluated using the gate set $\{\mathtt{H}, \mathtt{S}, \mathtt{CNOT}, \mathtt{T}\}$, widely considered a promising basis for universal fault-tolerant quantum computation. Since the T gate typically dominates the implementation cost due to hardware and error-correction constraints, we follow common practice and report the number of T gates (T), the T-depth (T-D), the number of qubits (W), and the T-$depth{\times}width$ metric. Finally, since the goal of the hybrid strategy is to show the effect of trading the exponential classical memory required by collision-based ISD algorithms for a quantum computation when subject to realistic constraints, we also imposed a maximum depth for a single quantum circuit of 2^{40} operations. Such depth represents "the approximate number of gates that presently envisioned quantum computing architectures are expected to serially perform in a year " [13].

Table 1: Comparison between the best classical ISD attack as per [6], and our hybrid Stern strategy. All values in $\log_2$.

Scheme	Sec. level	Classical				Hybrid Stern				
		$\log_2(M)$		$\sqrt{M}$						
		Op.s	M	Op.s	M	D_{sc}	T	T-D	W	T-D×W
HQC	L1	158	33	168	31	28	160	148	20	167
	L3	226	35	238	33	29	229	216	21	237
	L5	289	36	301	34	30	292	278	22	299
BIKE (key)	L1	166	31	176	30	27	169	157	19	176
	L3	231	41	242	32	28	234	221	21	241
	L5	300	45	311	33	29	303	289	22	310
BIKE (msg)	L1	158	32	168	30	27	161	149	19	168
	L3	224	34	235	32	28	227	214	21	234
	L5	290	61	301	33	28	293	279	22	300
Classic McEliece	L1	147	74	164	24	31	158	148	17	164
	L3	188	76	207	26	31	199	189	17	206
	L5	263	72	285	26	37	271	261	18	278
	L5	264	74	279	27	37	272	261	18	279
	L5	298	75	322	27	37	305	294	18	312

Tab. 1 shows that, when classical memory access is modeled with realistic sublinear costs such as $\sqrt{M}$, the hybrid approach yields an attack depth that is smaller than the corresponding number of classical bit operations by factors ranging from 2^{18} to 2^{26}. When evaluated comparing the quantum depth×width metric, the hybrid attack achieves performance comparable to classical approaches under the square-root memory model for nearly all considered cryptographic schemes. The only exception is the L5 parameter set of Classic McEliece, for which the hybrid approach provides a reduction by a factor between 2^7 and 2^{10}. Accounting for hardware constraints, quantum gates are expected to operate on nanosecond–microsecond timescales depending on the technology, compared to picosecond switching for classical gates. Incorporating this disparity, and even accounting for additional overheads due to quantum error correction, the hybrid approach may still retain a marked advantage over the best classical attacks in the passive quantum model, where circuit depth dominates the complexity

These observations underscore the importance of incorporating realistic memory-access costs and physical gate constraints when assessing post-quantum cryptosystems, and motivate further investigation of both fully-quantum and hybrid quantum–classical strategies for collision-based attacks. In particular, theoretical results [4, 5] indicate that collision search could potentially achieve speedups exceeding the quadratic improvements implied by standard quantum walk techniques; however, to date, no concrete circuit-level implementation has been proposed.

References

[1] Gorjan Alagic et al. *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*. NIST IR 8413. Gaithersburg, MD: National Institute of Standards and Technology, July 2022, p. 99. DOI: 10.6028/NIST.IR.8413.

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

[3] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. "On the Inherent Intractability of Certain Coding Problems (Corresp.)" In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 24.3 (1978), pp. 384–386. DOI: 10.1109/TIT.1978.1055873.

[4] Gilles Brassard, Peter Høyer, and Alain Tapp. "Quantum Cryptanalysis of Hash and Claw-Free Functions". In: *SIGACT News* 28.2 (1997), pp. 14–19. DOI: 10.1145/261342.261346.

[5] André Chailloux, María Naya-Plasencia, and André Schrottenloher. "An Efficient Quantum Collision Search Algorithm and Implications on Symmetric Cryptography". In: *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10625. Lecture Notes in Computer Science. Springer, 2017, pp. 211–240. DOI: 10.1007/978-3-319-70697-9_8.

[6] Andre Esser and Emanuele Bellini. "Syndrome Decoding Estimator". In: *Public-Key Cryptography - PKC 2022 - 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, March 8-11, 2022, Proceedings, Part I*. Ed. by Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe. Vol. 13177. Lecture Notes in Computer Science. Springer, 2022, pp. 112–141. DOI: 10.1007/978-3-030-97121-2_5.

[7] Andre Esser et al. "SoK: CryptographicEstimators - a Software Library for Cryptographic Hardness Estimation". In: *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security, ASIA CCS 2024, Singapore, July 1-5, 2024*. Ed. by Jianying Zhou et al. ACM, 2024. DOI: 10.1145/3634737.3645007.

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

[9] Giacomo Lancellotti et al. "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*. Ed. by Vivek De. ACM, 2024, 298:1–298:6. DOI: 10.1145/3649329.3657337.

[10] Giacomo Lancellotti et al. "Solving the Subset Sum Problem via Quantum Walk Search". In: *IEEE Transactions on Computers* (2025). DOI: 10.1109/TC.2025.3625044.

[11] P. J. Lee and E. F. Brickell. "An Observation on the Security of McEliece's Public-Key Cryptosystem". en. In: *Advances in Cryptology — EUROCRYPT '88*. Ed. by D. Barstow et al. Red. by G. Goos and J. Hartmanis. Vol. 330. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 275–280. DOI: 10.1007/3-540-45961-8_25.

[12] Frédéric Magniez et al. "Search via Quantum Walk". In: *SIAM Journal on Computing* 40.1 (2011), pp. 142–164. DOI: 10.1137/090745854.

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

[14] E. Prange. "The Use of Information Sets in Decoding Cyclic Codes". en. In: *IEEE Transactions on Information Theory* 8.5 (Sept. 1962), pp. 5–9. ISSN: 0018-9448. DOI: 10.1109/TIT.1962.1057777.

[15] Jacques Stern. "A Method for Finding Codewords of Small Weight". In: *Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings*. Ed. by Gérard D. Cohen and Jacques Wolfmann. Vol. 388. Lecture Notes in Computer Science. Springer, 1988, pp. 106–113. DOI: 10.1007/BFb0019850.