Decoding Cost Lens: Cost Estimation for Code-Based Cryptography

Tianrui Wang[1*] Anyu Wang[1] Yuxuan Li[2] Qicheng Teng[1] Bo Pang[3]

[1]Tsinghua University, Beijing, China
[2]Southern University of Science and Technology
[3]ICBC, Beijing, China

November 23, 2026

Abstract

We present Decoding Cost Lens, a platform for evaluating the concrete decoding costs of code-based cryptographic schemes. It maps key- and message-recovery objectives to explicit attack instances, integrates classical Hamming- and rank-metric estimators, and provides quantum estimates for binary Hamming instances. The evaluation accounts for known block structure and applicable multiple-target effects under stated cost conventions. We apply the platform to 64 parameter sets from 15 first-round candidate schemes in the NGCC call for next-generation commercial cryptographic algorithms, comparing estimated work factors with the specifications' claims and security targets. Four schemes have classical decoding estimates below both the corresponding declared cost and the target for at least one parameter set. Zero-direction maximal minors (ZD-MM) replaces one guessed zero with a search over directions in a single block. For BAG-Loong, the lowest ZD-MM costs are 113.1, 212.3, 299.1 and 386.7 bits in increasing parameter order. ZD-MM also puts key recovery below target for BAG-Piglet-128 and -256 (114.6 and 254.0 bits), and for all three BRA sets (112.1, 161.8 and 316.8 bits) after block transposition. For NSS-HQC, key and baseline message decoding costs fall below the 256-, 384-, and 512-bit classical targets; the corresponding baseline message costs are 226.0, 332.5, and 440.9 bits. NSS-HQC-128 also falls below its declared key-recovery cost while remaining above its target. The estimated quantum baseline message cost for NSS-HQC-512 is 251.6 bits against a 256-bit target.

*wangtr22@mails.tsinghua.edu.cn

Contents

1 Introduction **3**

2 Preliminaries **3**

3 Hamming Schemes **4**
3.1 Schemes and Decoding Instances 4
3.2 Classical Attack Algorithms and Costs 6
3.3 Quantum Attack Algorithms and Costs 7
3.4 Multiple Targets and Structural Assumptions 9
3.5 Evaluation Scope and Modeling Assumptions 11
3.6 Results 11

4 Rank Schemes **13**
4.1 Schemes and Decoding Instances 13
4.2 Classical Attack Algorithms and Costs 14
4.3 Structural Assumptions 15
4.4 Zero-direction maximal minors (ZD-MM) 16
4.5 Results 18

5 Conclusion **20**
5.1 Lower estimates below the target 20
5.2 Lower estimates that remain above the target 20
5.3 Estimates close to the specifications 21
5.4 Higher estimates under different cost models 22
5.5 Comparisons with security targets 22

A ZD-MM operation counts **23**

1 Introduction

The first-round public-key candidates in the Global Call for Next-Generation Commercial Cryptographic Algorithms (NGCC), published by the Institute of Commercial Cryptography Standards (ICCS), include code-based schemes assessed through Hamming-metric decoding, rank-metric decoding, and rank-support learning [1]. We use **Decoding Cost Lens (DCLens)** to recompute the decoding costs for 15 of these schemes. The evaluation covers 10 Hamming-metric schemes with 47 parameter sets and 5 rank-metric schemes with 17 parameter sets. For each scheme, we reconstruct the attack instances and compare the estimated costs with the declared values and security targets.

At concrete parameter sizes, decoding costs depend on the recovery objective, the available equations, the distribution of error weights or ranks, and the cost of linear algebra and list processing. Code structure can also provide related targets or known block information that changes the success probability. For each recovery route, we record the instance parameters and cost conventions before comparing the evaluated attacks. Classical costs are reported as $\log_2 T$, where T counts bit operations.

For the Hamming-metric schemes, we combine the finite-parameter information-set decoding models of CryptographicEstimators 2.1.1 [2] with a MAGE-Stern model [3]. We account for multiple targets and known block weights in the success probabilities and attack costs, and report key, message, and conditional combined-ciphertext (CT) costs separately. Quantum estimates for binary instances use finite-parameter adaptations of QPrange, QSS and QMMT+ [4].

For the rank-metric schemes, we reconstruct the classical combinatorial and algebraic attack formulas used in the specifications. The instances retain block lengths, block ranks, support intersections, and shared syndromes where applicable. Attack parameters are searched over specified ranges or selected for individual instances under the model constraints. Zero-direction maximal minors (ZD-MM) computes one kernel and reuses it to test all local directions for the final zero. For both ordinary MM and ZD-MM, we use $\omega = 2.8$ for BAG-Loong and BAG-Piglet and $\omega = 2$ for BRA, BRQC and C-Multi-UR-AG.

We find that many recomputed costs closely match the values reported in the specifications, while NSS-HQC and BAG-Loong show larger discrepancies. For NSS-HQC, the differences affect both classical key recovery and baseline message recovery, with estimates below the stated targets at the 256-, 384-, and 512-bit levels. For BAG-Loong, the recomputed maximal-minors costs fall below the classical targets at all four parameter levels. ZD-MM also puts key recovery below the classical targets for BAG-Piglet-128, BAG-Piglet-256 and all three BRA parameter sets. The final comparison examines both the agreement between recomputed and declared work factors and whether the resulting estimates meet the stated security targets.

Section 2 introduces the preliminaries for code theory. Sections 3 and 4 present the attack instances, cost models, and results for Hamming-metric and rank-metric schemes, respectively. Section 5 compares the estimates with the declared costs and security targets.

2 Preliminaries

Hamming metric and linear codes. For a vector $\boldsymbol{x} \in \mathbb{F}_q^N$, its Hamming weight $\mathrm{wt}_H(\boldsymbol{x})$ is the number of nonzero coordinates; the Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ is $\mathrm{wt}_H(\boldsymbol{x} - \boldsymbol{y})$. A linear $[N, K]_q$ code is a K-dimensional subspace of $\mathbb{F}_q^N$. It can be described by a full-row-rank parity-check matrix $H \in \mathbb{F}_q^{(N-K)\times N}$ as

$$\mathcal{C} = \{\boldsymbol{c} \in \mathbb{F}_q^N : H\boldsymbol{c}^\mathsf{T} = \boldsymbol{0}\}.$$

If a codeword $\boldsymbol{c}$ is perturbed by an error $\boldsymbol{e}$, the received vector $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ has syndrome $\boldsymbol{s} = H\boldsymbol{y}^\mathsf{T} = H\boldsymbol{e}^\mathsf{T}$. The syndrome therefore provides linear constraints on the error without requiring knowledge of the transmitted codeword [5].

Hamming-metric syndrome decoding. Given H, a syndrome $\boldsymbol{s} \in \mathbb{F}_q^{N-K}$, and a prescribed positive weight W, the Hamming-metric syndrome decoding problem (SD) is to find $\boldsymbol{e} \in \mathbb{F}_q^N$ such that

$$H\boldsymbol{e}^{\mathsf{T}} = \boldsymbol{s}, \qquad \mathrm{wt}_H(\boldsymbol{e}) = W.$$

When $\boldsymbol{s} = \boldsymbol{0}$, the same equations ask for a codeword of weight W in $\mathcal{C}$. This is low-weight codeword finding; the positive weight excludes the zero vector. Both problems are described by $(N, K, W; q)$ and use exact weight rather than an upper bound.

Rank metric and support Let $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)$ a basis of $\mathbb{F}_{q^m}/\mathbb{F}_q$. $\boldsymbol{x}$ can be represented by an $m \times n$ matrix $\boldsymbol{M_x}$ such that its i^{th} column represents the coordinates of x_i in the basis $\boldsymbol{\beta}$

$$\boldsymbol{x} \leftrightarrow \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix}$$

with $x_j = \sum_{i=1}^{m} x_{ij}\beta_i$ for all $j \in [1..n]$.

The rank weight of $\boldsymbol{x}$ is defined by $\mathrm{wt}_R(\boldsymbol{x}) \stackrel{\text{def}}{=} \mathrm{rank}\, \boldsymbol{M_x}$. Similarly, the rank distance of $\boldsymbol{x}$ and $\boldsymbol{y}$ is $d_R(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{wt}_R(\boldsymbol{x} - \boldsymbol{y})$. These definitions do not depend on the choice of the basis.

The support of $\boldsymbol{x}$ is the $\mathbb{F}_q$-subspace of $\mathbb{F}_{q^m}$ generated by the coordinates of $\boldsymbol{x}$.

Rank syndrome decoding (RSD) Let $\boldsymbol{H}$ be the parity-check matrix of an $[n, k]$ code, $\boldsymbol{s} \in \mathbb{F}_{q^m}^{n-k}$ and w an integer. The RSD problem consists to find $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ such that

$$\boldsymbol{H}\boldsymbol{e}^T = \boldsymbol{s}, \qquad \mathrm{wt}_R(\boldsymbol{e}) = w$$

3 Hamming Schemes

3.1 Schemes and Decoding Instances

The assessment covers 47 parameter sets from 10 schemes in the first-round candidate list published by ICCS [1]. Each decoding instance is specified by (N, K, W): the length of the unknown vector, $K = N - \mathrm{rank}(H)$, and its total Hamming weight. Here K refers to the attack equations, not necessarily to the scheme's message-encoding code. All instances are binary except UVW-KEM, which uses $q \in \{433, 857, 1709\}$.

In the tables, r is the cyclic block length and d is Mito's number of ring coordinates, a power of two. Weight subscripts identify the corresponding secret or ephemeral blocks. Target multiplicities are treated separately in Section 3.4.

3.1.1 Baseline Instances

Table 1 lists the key- and message-recovery instances derived from the scheme specifications [6–15]. Key recovery for BIKE-MLThre and TRIKE searches for a nonzero codeword satisfying the public homogeneous relation; the other key models solve syndrome equations for sparse secret vectors. For HEP-QC, these are (x, y) and do not include the secret coding matrices T, P. For HARE, HEP-QC, NSS-HQC, TriQ-KEM, Mito, and QUBE, the baseline message model recovers ephemeral vectors using only the first ciphertext component u. The remaining schemes target the ciphertext error vector. Subsequent public-code decoding, message reconstruction, and candidate verification are treated as postprocessing and are not included in these decoding costs.

Table 1: Scheme coverage and baseline attack parameters (N, K, W).

Scheme	Sets	Key recovery	Message recovery
BIKE-MLThre	4	$(2r, r, w)$	$(2r, r, t)$
TRIKE	5	$(3r, 2r, w)$	$(3r, r, t)$
HARE	6	$(2r, r, 2w)$	$(2r, r, 2w_r)$
HEP-QC	4	$(2r, r, 2w)$	$(2r, r, 2w_r)$
NSS-HQC	4	$(2r, r, 2w_{\mathrm{sk}})$	$(2r, r, w_{r_1} + w_{r_2})$
TriQ-KEM	4	$(2r, r, w_x + w_y)$	$(2r, r, 2w_r)$
Mito	9	$(2dr, dr, W_{\mathrm{key}})$	$(2dr, dr, W_u)$
QUBE	5	$(3r, r, 2w_x + w_y)$	$(3r, 2r, 2w_f + w_g)$
QCTM	3	—	$(n, n - R_Q, W_Q)$
UVW-KEM	3	—	$(n, k_1 + k_2, w)$

For Mito, $W_{\mathrm{key}} = \sum_i (w_{x,i} + w_{y,i})$ and $W_u = \sum_i (w_{r,i} + w_{t,i})$. For QCTM, $R_Q = 18(t-1) + 1$ and $W_Q = (t-1)/2$, where t is the Goppa degree. BIKE-MLThre and TRIKE use total weights w and t, whereas w in HARE and HEP-QC denotes one secret-block weight. A dash indicates that no key-recovery instance is considered. TRIKE uses the five parameter sets in the updated specification v1.6.

3.1.2 Combined Ciphertext Instances

For HARE, HEP-QC, NSS-HQC, and TriQ-KEM, the ciphertext (u, v) has the schematic form

$$\begin{aligned} u &= r_1 + hr_2, \\ v &= \mathrm{Comp}\big(\mathrm{Encode}(\mu) + \pi(sr_2 + e)\big). \end{aligned}$$

Here μ is the message, π selects the retained coordinates, and Comp denotes public compression or processing (the identity when absent).

The combined model takes (u, z) as input, where $z = sr_2 + e$ is the full-length masking term in the second component. It assumes that z is known and excludes its recovery cost. Table 2 gives the resulting parameters. For Mito, the second relation selects one ring coordinate, so the additional error block has length r. The QUBE instance is derived analogously from its two ciphertext equations.

Table 2: Conditional message-recovery parameters using both ciphertext equations.

Scheme	(N, K, W)
HARE, HEP-QC, TriQ-KEM	$(3r, r, 2w_r + w_e)$
NSS-HQC	$(3r, r, w_{r_1} + w_{r_2} + w_e)$
Mito	$((2d+1)r, dr, W_u + w_e)$
QUBE	$(4r, 2r, 2w_f + w_g + w_e)$

3.1.3 Mito's Public Parity Constraints

Mito's secret vectors x and y have fixed block weights, so their block parities are public [12]. Appending the d parities of y to the public key equation adds d independent constraints, while the parities of x are already implied. This gives an additional key-recovery instance

$$(N, K, W) = (2dr, dr - d, W_{\text{key}}).$$

It is evaluated using generic SD costs at the reduced dimension, with no additional cost reduction for the parity constraints. Block-weight conditioning is not combined with this rank model.

3.2 Classical Attack Algorithms and Costs

We compare classical information-set decoding algorithms using the finite-parameter models in CryptographicEstimators (CE) 2.1.1 [2], together with a MAGE-Stern model. Time is measured in bit operations and memory in bits; reported costs use base-two logarithms. Memory access is assigned constant cost, and a specified memory limit restricts the admissible algorithm parameters.

3.2.1 Information-Set Decoding

An information set is a set of K coordinates whose values determine a codeword uniquely. Equivalently, the complementary $N - K$ columns of H form an invertible matrix. Information-set decoding (ISD) repeatedly chooses a coordinate partition, simplifies the syndrome equations by Gaussian elimination, and searches for error patterns with a prescribed weight distribution. Each partial candidate is completed using the remaining equations and accepted if its total weight is W [4].

All variants balance the work in one trial against its probability of success:

$$T_{\text{C}} \approx C_{\text{pre}} + \frac{C_{\text{iter}}}{P_{\text{succ}}},$$

where C_{pre} is reusable preprocessing, C_{iter} includes elimination and candidate search, and P_{succ} is the modeled success probability. The main algorithmic differences lie in how candidates are constructed and matched.

Prange assumes an error-free information set and solves directly for the remaining error coordinates [16]. Lee–Brickell allows a small number of errors on the information set and enumerates their possible values before completing the candidate [17].

Stern replaces direct enumeration with a two-list search, matching partial errors through their syndrome contributions [18]. Dumer combines two-list matching with partial elimination, enlarging the region searched for errors [19]. Ball-collision decoding extends the two-list approach by allowing errors in additional coordinate blocks used for matching [20].

BJMM uses multiple representations of an error vector and merges partial candidates through several list levels [21]. The assessment includes depths two and three, together with BJMMplus, BJMMdw, and BJMMpdw, which modify the time–memory tradeoff or the weight distribution among lists [22, 23]. May–Ozerov and Both–May use nearest-neighbor searches to find list pairs whose sum has the required small weight [24, 25].

The binary comparison includes Prange and the list-based families above; UVW-KEM uses the q-ary Prange, Stern, and Lee–Brickell models. Nonbinary costs include the choice of nonzero field values as well as error positions [26]. For each algorithm, finite combinatorial counts and operation costs are used to select the weight distribution, list sizes, and matching parameters under the memory limit.

3.2.2 MAGE-Stern

Multilevel Amortized Gaussian Elimination (MAGE) reduces repeated elimination work by organizing related information sets into a tree [3]. Each internal node performs part of the elimination, which its descendants reuse; each leaf completes a Stern search. The tree depth and branching factors control the tradeoff between shared elimination and the number of searches performed. A recurrence tracks the probability that at least one leaf reaches a suitable error distribution, accounting for the shared choices along tree paths.

For ν processed syndromes, let $C_{\text{tree}}(\nu)$ include the tree's elimination, list, and per-syndrome transformation costs. The total time estimate is

$$T_{\text{MAGE}} = T_0(\nu) + \frac{C_{\text{tree}}(\nu)}{P_{\text{tree}}}, \qquad T_0(\nu) = (N+\nu)(N-K)^2.$$

Here T_0 is the initial conversion cost and P_{tree} is the modeled probability that a tree succeeds. Costs use CE's bit-operation convention, with N bits of storage per list record. For SD and its structured syndrome variants, this model assumes one acceptable solution per syndrome. The treatment of multiple targets is described in Section 3.4.

3.3 Quantum Attack Algorithms and Costs

The quantum assessment includes QPrange, QSS, and QMMT+. QPrange applies amplitude amplification to Prange's information-set search. QSS and QMMT+ use the framework of Kachigar and Tillich [4], combining a four-list search with amplitude amplification and a quantum walk. QMMT+ additionally uses multiple representations obtained through binary cancellations. The estimates include elimination and checking or list-processing costs. QSS and QMMT+ parameters are optimized independently of the classical algorithms, whereas QPrange uses a fixed parameter choice. The following models adapt these algorithms to finite parameters. UVW-KEM is assessed only with classical algorithms.

3.3.1 QPrange

QPrange searches for a partition whose K information coordinates contain no errors, solves for the remaining coordinates, and checks the candidate. Writing $m = N - K$, the hit probability for one solution under uniform coordinate selection is $P = \binom{m}{W}/\binom{N}{W}$. The rank-success factor ρ is evaluated using a random binary matrix model, giving an effective success probability ρP. The resulting time estimate is

$$T_{\text{QPrange}} = N(m+1) + \frac{N(G_{\text{Q}}+1)}{\sqrt{\rho P}},$$

where G_{Q} is the preparation and elimination cost in vector operations, and the direct check costs one vector operation. The factor $1/\sqrt{\rho P}$ accounts for amplitude amplification; the one-time cost $N(m+1)$ is measured in bit operations. Known block weights can change P as described in Section 3.4.3.

3.3.2 QSS and QMMT+ List Structures

Partial Gaussian elimination leaves ℓ equations on a search region of $K_\ell = K + \ell$ coordinates. The remaining $N - K - \ell$ coordinates are determined once a candidate for this region is chosen. Let p be the number of errors in the search region, so the remaining region must contain $W - p$ errors. QPrange corresponds to the direct-check branch at $\ell = p = 0$, with G_{Q} evaluated at $\ell = 0$.

QSS divides the search region into four equal parts and requires $p/4$ errors in each part. Each of its four initial lists enumerates the corresponding partial vectors, giving list size $V_{\mathrm{QSS}} = \binom{K_\ell/4}{p/4}$. A target with this weight distribution has a unique decomposition across the four parts, so its representation count is $R = 1$.

QMMT+ divides the search region into two halves, each containing $p/2$ target errors. Each half contributes two initial lists. Their partial vectors can share nonzero positions that cancel under binary addition, allowing several decompositions of the same target. With $c/2$ cancellation positions in each half, the initial-list size and representation count are

$$V_{\mathrm{QMMT+}} = \binom{K_\ell/2}{p/4 + c/2}, \qquad R_{\mathrm{QMMT+}} = \left[\binom{p/2}{p/4}\binom{K_\ell/2 - p/2}{c/2}\right]^2.$$

Here c is even and $0 \le c \le K_\ell - p$. The balanced splits require $4 \mid p$, with $4 \mid K_\ell$ for QSS and even K_ℓ for QMMT+. Setting $c = 0$ gives ordinary QMMT. The cancellation parameter controls the tradeoff between the representation count and the initial-list size. In both algorithms, list pairs are merged in two stages to find a combination satisfying the retained syndrome equations.

3.3.3 Success Probability and Time Complexity

Let P be the probability that a uniformly placed weight-W support has the balanced distribution required above. Partial elimination must also succeed; its rank-success factor ρ is evaluated using a random binary matrix model. Let L denote the number of acceptable solutions to one syndrome equation, distinct from the number R of representations of a single solution.

Additional filtering reduces the lists before their final merge. The model uses a filter bits whose values are searched by amplitude amplification, and b filter bits whose loss is offset by having several representations. It searches $a \ge 0$, $0 \le b \le \lfloor \log_2 R \rfloor$, and $a + b \le \ell$. The representation-survival factor and effective success probability are modeled as

$$\sigma = \frac{R}{R + 2^b - 1}, \qquad P_{\mathrm{eff}} = \rho[1 - (1 - P\sigma)^L].$$

QSS takes $b = 0$ and $\sigma = 1$. The formula treats solution hits as independent once the common rank condition holds.

Let G_{Q} be the elimination cost and $\mathcal{Q}$ the list-search cost, both measured in vector operations. Charging N bit operations per vector operation gives the quantum time estimate

$$T_{\mathrm{Q}} = T_0 + \frac{N(G_{\mathrm{Q}} + 2^{a/2}\mathcal{Q})}{\sqrt{P_{\mathrm{eff}}}}.$$

The preprocessing cost T_0 is measured in bit operations. The outer amplitude amplification contributes $1/\sqrt{P_{\mathrm{eff}}}$, while $2^{a/2}$ accounts for the inner filter search. Elimination is repeated in the outer search, so its cost remains inside that repetition factor.

The list search compares constructing the complete lists with a quantum walk on smaller subsets. The walk reuses its lists by replacing entries, paying for initial construction, updates, and candidate checks [27]. These costs include ordered list merges and index maintenance. A separate direct check handles the cases in which the search region is all zero or all one, without forming lists; in that case $a = 0$ and $\mathcal{Q}$ is replaced by the checking cost.

With $m = N - K$ as above, the default partial-elimination cost is

$$G_{\mathrm{Q}} = m(m - \ell) + m\lceil \log_2(N + 2) \rceil^2 + N.$$

An alternative uses partial M4RI elimination, which groups pivots into blocks and precomputes row combinations to reduce update work [28]. Its block width is optimized over the combined cost of table construction and row updates. Both choices assume coherent data access and reversible processing. Quantum costs are specified at the operation level; memory and circuit-level costs are outside the scope of the evaluation (Section 3.5).

3.4 Multiple Targets and Structural Assumptions

3.4.1 Available Target Orbits

In quasi-cyclic syndrome decoding (QCSD), simultaneous cyclic shifts of the unknown blocks give related syndrome equations under the same matrix. Solving any one of these equations allows the solution to be shifted back. This leads to decoding one out of many (DOOM), where an attack seeks a solution to any of M available syndromes [29]. Sharing elimination and part of the list work amortizes their cost across syndromes. The scheme mappings use one target solution per syndrome, $L = 1$.

The cyclic models use up to $M = r$ targets. Mito additionally has dyadic symmetry, giving quasi-dyadic syndrome decoding (QDSD); its key and first-component models use $M = dr$. Its combined model uses only the cyclic shifts, so $M = r$. QCTM uses its order-19 automorphism to obtain $M = 19$ targets [14], while UVW-KEM uses $M = 1$.

For BIKE-MLThre and TRIKE key recovery, shifting the secret leaves the syndrome equal to zero. This is quasi-cyclic codeword finding (QCCF): here M counts acceptable nonzero codewords in one homogeneous system, rather than different syndromes. Multiple codewords and multiple syndromes are therefore handled separately in the cost models.

The stated counts assume that the full orbit gives distinct, usable targets. For Mito's parity model, the block-parity labels must be permuted along with the syndrome. Truncated or compressed ciphertexts may provide fewer usable targets, and the statistical dependence between shifted targets is not modeled.

3.4.2 Algorithm-Specific Multiple-Target Costs

Processing more syndromes increases both the chance of success and the work per trial. The binary Prange, Stern, Dumer, and BJMM models therefore select the processed syndrome count $1 \le \nu \le M$ together with their other parameters. Let P_0 be the single-solution hit probability, and let G_{C}, W_{shared}, and W_{target} be the elimination, shared-list, and per-syndrome list costs in vector operations. With $m = N - K$, the finite model uses the clipped linear success approximation

$$P_{\mathrm{succ}} = \min(1, \nu L P_0), \qquad T_{\mathrm{C}}(\nu) = \frac{N[G_{\mathrm{C}} + J(\nu)(W_{\mathrm{shared}} + \nu W_{\mathrm{target}})] + (\nu - 1)m^2}{P_{\mathrm{succ}}}.$$

G_{C} includes processing the first syndrome; each additional one requires m^2 bit operations and m bits of storage. For Stern, $J(\nu)$ counts projection retries at the selected target count; elimination is not repeated for each projection. The other models use $J(\nu) = 1$, with BJMM's representation retries included in the list costs. Stern and Dumer share one list across syndromes and enlarge the other with syndrome-specific entries. The BJMM multiple-syndrome model uses depth two and includes targets in one of its four initial lists [30]. The single-syndrome CE estimates remain candidates in the comparison. Other binary CE algorithms use a simpler DOOM-inspired approximation: their time exponent is reduced by $\frac{1}{2}\log_2 M$, with a floor of zero, while their parameters and memory estimates stay unchanged. BJMM also compares an independently optimized depth-three CE candidate with this square-root discount against its explicit depth-two

DOOM candidate. Without block conditioning, the explicit models retain CE's finite list counts, rounding, and memory budgets, and agree pointwise with CE at one target in their common parameter domain.

For MAGE-Stern, let S be the probability that one tree finds a specified target. Assuming independent target hits, the multiple-syndrome model uses $P_{\text{tree}} = 1 - (1 - S)^{\nu}$ and includes the work for each syndrome in the tree cost. For QCCF, it processes only the zero syndrome and uses $1 - (1 - S_{\text{cw}})^{M}$, where S_{cw} is the modeled probability of finding one specified codeword. The codeword count changes the success probability without multiplying the inner list work.

The CE and QSS/QMMT+ QCCF models likewise treat the M codewords as solutions to one equation. CE uses this count before parameter optimization, and the quantum model sets $L = M$ in P_{eff}. No further target-count discount is added. For QCSD and QDSD, the quantum assessment uses a single-syndrome estimate and does not apply a multi-syndrome speedup.

Codeword finding also requires excluding the zero vector. MAGE-Stern requires nonzero information-coordinate patterns, and the quantum model excludes its all-zero direct check. The generic CE and quantum models do not fully account for the cost of removing trivial zero-syndrome matches. In particular, Prange's zero-information-vector step returns the zero codeword for an invertible elimination block, so its generic SD estimate is not a valid nonzero-codeword attack cost. QPrange therefore does not support nonzero-codeword QCCF.

3.4.3 Known Block Weights

For binary instances with known block lengths n_i and weights w_i, where $\sum_i n_i = N$ and $\sum_i w_i = W$, the model can bias coordinate selection toward the prescribed weight distribution. If the outer tail contains R_i coordinates from block i, the probability of p errors outside that tail is

$$P_{\boldsymbol{R}}(p) = [z^p] \prod_i \left(\sum_j \frac{\binom{n_i - R_i}{j}\binom{R_i}{w_i - j}}{\binom{n_i}{w_i}} z^j \right).$$

The total tail size is fixed by the algorithm parameters. The integer quotas R_i are optimized and compared with uniform coordinate selection. The success probabilities are computed from these finite hypergeometric distributions.

For CE and explicit DOOM models, this probability replaces the aggregate outer weight event before computing repetitions and target gains. Uniform shuffling within the selected regions retains the original conditional split factors and inner list, representation, elimination, and memory costs. For QPrange, QSS and QMMT+, it updates P in P_{eff}; the rank factor and inner costs remain unchanged. MAGE-Stern instead uses the block-weight distribution in the first pivot transition of its tree recurrence, then shuffles the remaining coordinates uniformly.

For example, the conditional HARE combined instance has block lengths (r, r, r) and weights (w_r, w_r, w_e). For QPrange, the model uses this profile to optimize the outer-tail allocation in $P_{\boldsymbol{R}}(0)$.

The scheme mappings supply fixed block weights where available; BIKE-MLThre and TRIKE message errors have only a total-weight constraint, and QCTM and UVW-KEM use the generic model. For Mito, dyadic shifts can permute unequal block weights. The comparison therefore includes both the targets preserving the chosen weight profile and the full orbit with uniform selection. Its parity-rank model remains a separate candidate. Prange's outer allocation uses discrete concave optimization; other large allocation spaces use a bounded integer search, retaining the generic candidate.

3.5 Evaluation Scope and Modeling Assumptions

Attack scope. The evaluation estimates decoding costs, including internal candidate checks. It excludes subsequent key or message reconstruction, scheme-level verification, structural attacks beyond the listed routes, decoding-failure attacks, and implementation attacks. The conditional CT models (Table 2) exclude the cost of recovering the full masking term, including elimination of the unknown message and handling of truncation or compression. Their application to public ciphertexts requires a reduction preserving the assumed equations, error distribution, and usable targets.

Treatment of structure. The models incorporate attack dimensions, block weights, and target orbits. We model supports as uniform within each prescribed fixed-weight block; bounded-density restrictions and additional support correlations are omitted. At fixed attack parameters, block conditioning changes only the coordinate-selection success probability (Section 3.4.3); list sizes and per-trial costs can change through parameter reoptimization. No additional structural speedup is assumed, and gains from different constraints are combined only when their joint effect is modeled.

Statistical assumptions. Apart from the structure above, matrix-rank, syndrome-label, and representation statistics follow generic random-code approximations. The quantum rank factor is evaluated without conditioning on the chosen block allocation. Target success probabilities use the approximations in Section 3.4, which assume distinct, usable targets and omit correlations between cyclic shifts. Quantum list updates use average collision loads, with no separate allowance for exceptional loads and restarts. The resulting estimates do not establish runtime bounds for individual structured keys or the scheme's key distribution.

Cost conventions. Binary vector operations cost N bit operations, and classical list records occupy N bits. Nonbinary field additions and stored field elements cost $\log_2 q$ bit operations and bits, respectively. The assessment imposes no classical memory bound and assumes constant-cost memory access; list construction, matching, and index maintenance are included. Reusable preprocessing is counted once; elimination and list costs that depend on the trial are counted at each repetition. Quantum costs assume coherent access and reversible processing. The evaluation excludes complete circuit gate counts, quantum memory costs, and physical error-correction overhead, and imposes no quantum memory bound.

3.6 Results

The assessment covers 47 parameter sets and 135 routes, with computed classical and quantum costs for 129 and 126 routes, respectively. Tables 3 and 4 give the minimum estimated $\log_2 T$ for each objective and route category, separately for classical (C) and quantum (Q) models. Known block weights and available target orbits are used as described above, with no classical memory bound. All reported quantum minima use QMMT+ with the default partial-elimination cost model.

Key columns minimize over the available key routes. Message columns use the first-component model where available and the direct error-recovery model otherwise. CT denotes the conditional combined-ciphertext route, with the additional assumptions of Table 2. Bold type marks the lowest evaluated cost in each row, comparing Key, Message, and CT separately for classical and quantum costs. Minima are selected using the original three-decimal results before rounding; ties at that precision are all marked. A dash denotes an unimplemented key model or an absent CT route; "n/a" denotes unsupported nonbinary quantum estimation. QPrange is included in the HARE and HEP-QC results.

Table 3: Minimum decoding costs $\log_2 T$: BIKE-MLThre through TriQ-KEM.

	Key		Message		CT	
Parameter set	C	Q	C	Q	C	Q
BIKE-MLThre-128	146.8	**95.2**	**145.9**	98.2	—	—
BIKE-MLThre-192	**210.3**	**127.9**	211.0	131.9	—	—
BIKE-MLThre-256	277.9	**162.4**	**275.9**	165.3	—	—
BIKE-MLThre-512	549.1	300.0	**536.1**	**297.8**	—	—
TRIKE-1	133.8	**87.7**	**131.3**	90.8	—	—
TRIKE-3	**200.0**	**122.2**	200.0	126.3	—	—
TRIKE-5	266.3	**156.5**	**264.4**	159.4	—	—
TRIKE-7	398.2	**224.2**	**398.1**	227.5	—	—
TRIKE-9	530.9	**291.1**	**525.6**	292.1	—	—
HARE-1	144.6	**98.2**	144.6	**98.2**	**144.1**	98.5
HARE-2	**170.6**	**111.9**	**170.6**	**111.9**	171.8	112.4
HARE-3	212.9	133.7	212.9	133.7	**211.8**	**133.4**
HARE-5	275.1	165.5	275.1	165.5	**274.4**	**165.3**
HARE-7	**399.3**	**228.7**	**399.3**	**228.7**	399.7	229.1
HARE-9	531.1	295.6	531.1	295.6	**528.9**	**294.7**
HEP-QC-1	**144.8**	**98.6**	162.4	107.3	146.4	100.2
HEP-QC-3	**213.1**	**134.1**	240.5	147.4	215.0	135.5
HEP-QC-5	**275.3**	**165.9**	310.6	183.2	276.6	167.0
HEP-QC-7	**535.8**	**298.1**	606.4	333.4	536.9	299.1
NSS-HQC-128	159.8	107.4	**146.2**	**100.7**	147.0	102.0
NSS-HQC-256	247.6	152.2	**226.0**	**141.8**	226.2	142.1
NSS-HQC-384	364.2	212.0	332.5	196.5	**332.0**	**196.4**
NSS-HQC-512	480.6	271.1	440.9	251.6	**439.6**	**251.2**
TriQ-KEM-128	146.6	**99.3**	146.6	**99.3**	**145.9**	99.4
TriQ-KEM-256	277.0	**166.3**	277.0	**166.3**	**277.0**	166.5
TriQ-KEM-384	404.6	231.3	404.6	231.3	**404.1**	**231.2**
TriQ-KEM-512	532.5	**296.1**	532.5	**296.1**	**532.1**	296.1

Table 4: Minimum decoding costs $\log_2 T$: Mito through UVW-KEM.

	Key		Message		CT	
Parameter set	C	Q	C	Q	C	Q
Mito-1-128	**185.1**	**119.9**	189.2	122.1	190.2	122.4
Mito-1-256	**282.4**	**169.8**	287.1	172.1	287.9	172.4
Mito-1-512	**540.4**	**300.4**	545.6	302.9	546.0	303.2
Mito-1-E-128	**185.0**	**119.7**	189.1	121.8	190.1	122.1
Mito-1-E-256	**282.3**	**169.5**	287.0	171.8	287.8	172.1
Mito-1-E-512	**540.3**	**300.2**	545.5	302.8	545.9	303.1
Mito-2-E-128	**199.7**	**127.7**	203.6	129.8	203.5	129.7
Mito-2-E-256	**297.3**	**177.8**	301.6	180.0	302.4	180.1
Mito-2-E-512	**557.7**	**309.5**	562.3	311.8	562.7	311.9
QUBE-128	**145.1**	**98.5**	146.8	98.6	147.8	100.9
QUBE-192	**209.0**	**131.3**	210.3	131.9	211.3	133.6
QUBE-256	**273.5**	**164.4**	275.4	165.6	276.5	166.9
QUBE-384	**402.5**	**230.1**	405.6	232.4	406.5	232.9
QUBE-512	**530.4**	**295.0**	534.0	297.5	534.6	297.9
QCTM-128	—	—	**150.9**	**96.7**	—	—
QCTM-256	—	—	**272.9**	**160.9**	—	—
QCTM-512	—	—	**530.3**	**297.2**	—	—
UVW-KEM-128	—	—	**146.5**	n/a	—	—
UVW-KEM-256	—	—	**275.9**	n/a	—	—
UVW-KEM-512	—	—	**533.3**	n/a	—	—

4 Rank Schemes

4.1 Schemes and Decoding Instances

The assessment covers 17 parameter sets from five rank-metric KEMs in the first-round candidate list [1]: BAG-Loong, BAG-Piglet, BRA, BRQC, and C-Multi-UR-AG. All use $\mathbb{F}_{q^m}$ with $q = 2$. For each parameter set we evaluate a key-recovery instance derived from the public relation and a message-recovery instance derived from the ciphertext relations. The costs cover recovery of the low-rank vector or support.

An instance is specified by the extension degree m, block lengths $(n_1, \ldots, n_\ell)$, block ranks $(r_1, \ldots, r_\ell)$, and dimension k of the code in the attack equations. The rank of a block is the dimension of its support over $\mathbb{F}_q$. When the supports are disjoint, the joint rank is $r = \sum_i r_i$. For two supports intersecting in dimension g, the rank is $r_1 + r_2 - g$. Table 5 summarizes the mappings. A single parameter n_0 denotes the common block length, and N_1, N_2 denote the numbers of syndromes in the two C-Multi-UR-AG instances.

Table 5: Key- and message-recovery instances for the original attacks. Lengths are listed by block, and k is the code dimension in the attack instance.

Scheme	Sets	Key instance	Message instance	Support structure
BAG-Loong	4	(n_0, n_0), $k = n_0$	$(n_0, n_0 + n_1)$, $k = n_0$	Two supports intersect in dimension g
BAG-Piglet	4	Three blocks of length n_0, $k = 2n_0$	Four blocks of length n_0, $k = 2n_0$	Disjoint block supports
BRA	3	Two blocks of length n_0, $k = n_0$	Three blocks of length n_0, $k = n_0$	Disjoint block supports
BRQC	3	Two blocks of length n_0, $k = n_0$	Three blocks of length n_0, $k = n_0$	Disjoint block supports
C-Multi-UR-AG	3	$(2n_0, n_0, w_1)$, N_1 syndromes	$(2n_0 + N_1, n_0, w_2)$, N_2 syndromes	Shared support

In the final row (n, k, w) gives the single-block rank-decoding parameters used as the basis for the multi-syndrome estimates. For example, BAG-Loong-128 uses block ranks $(5, 5)$ with $g = 4$ in the key instance, giving a joint rank of 6, while its message instance uses $g = 3$. The second ciphertext component contains an encoded message. Section 4.4 gives the public ciphertext relations used for the structured MM estimates.

4.2 Classical Attack Algorithms and Costs

The classical estimates compare combinatorial support or coefficient enumeration with algebraic linearization. Each algorithm combines a guessing cost with the cost of solving the resulting system. Costs are $\log_2 T$ in bit operations, with $\omega = 2.8$ for BAG-Loong and BAG-Piglet and $\omega = 2$ for the other schemes. We assume one solution per instance and apply no explicit memory bound. For the original BRA and BRQC attacks, we use the submitted blockwise MM parameters, with revised block orders or guessing and puncturing parameters for four message-recovery tuples that fail the equation-count threshold. The other MM estimators search their parameter ranges.

4.2.1 Combinatorial Attacks

OJ. The coefficient-side OJ attack enumerates one side of a rank factorization and solves for the other. Its blockwise cost uses the individual block lengths and ranks. The enumeration exponent depends on the block containing the relevant position in the extended code. For BAG-Loong the implementation compares the two block orders. With lengths d_1, d_2 and joint rank $t = r_1 + r_2 - g$, its cost for one ordering is

$$T_{\mathrm{OJ}} = \big(m(t-1) + d_1 - r_1\big)^{\omega} q^{(d_1 - r_1)(r_1 - 1) + r_2}.$$

Thus the overlap enters the polynomial factor through t in this expression.

AGHT. The attack guesses a superspace of the error support. Every nonzero scalar multiple αe, $\alpha \in \mathbb{F}_{q^m}$, is also a codeword of the extended code, giving the $-m$ exponent in the first candidate below:

$$T_{\mathrm{AGHT}} = \min\Big\{ \big((n-k-1)m\big)^{\omega} q^{r\lceil (k+1)m/n \rceil - m}, \quad m^2 (n-k)^{\omega} q^{rk} \Big\}.$$

Here $n = \sum_i n_i$ and $r = \sum_i r_i$ for the disjoint-block instances. The implementation uses these totals in AGHT.

PRR. For each block, PRR guesses a subspace of dimension u_i containing the support and then uses erasure decoding. For BRA and BRQC, the column-support candidate has cost

$$T_{\mathrm{PRR-S}} = \big(m(n-k-1)\big)^{\omega} q^{\sum_i r_i (m - u_i) - m}, \quad r_i \le u_i \le m, \quad \sum_i u_i \le m, \quad \sum_i n_i u_i \le m(n-k-1).$$

The row-support candidate costs

$$T_{\mathrm{PRR-C}} = m^2(n-k)^\omega q^{\sum_i r_i(n_i-u_i)}, \quad r_i \le u_i \le n_i, \quad \sum_i u_i \le n-k.$$

The estimator allocates the dimension budget across blocks and selects the cheaper candidate. BAG-Loong and BAG-Piglet use their own block constraints and cost factors.

4.2.2 Algebraic Attacks

MM. Maximal-minors linearization converts rank constraints into a linear system. For BRA and BRQC, guessing a_i coordinates and puncturing p_i in block i gives the unknown count and modeled cost

$$U = \prod_i \binom{n_i - a_i - p_i}{r_i}, \qquad T_{\mathrm{MM}} = q^{\sum_i a_i r_i} E U^{\omega-1}.$$

The equation count E uses the blockwise extended-Gabidulin minor count [31]. Candidate parameters are evaluated at the threshold $E \ge U - 1$. The other schemes use specialized counts, including the overlapping-block count described below. For C-Multi-UR-AG, the single-block MM model searches the number of guessed coordinates.

SM. Support-minors linearization multiplies equations by monomials of degree b and solves the resulting system. C-Multi-UR-AG compares degrees $b \in \{1,2,3\}$, guessed-coordinate counts $a \le 80$ and puncture counts $p \le 2$. Its cost accounts for the numbers of equations and unknowns and the m^2 factor in higher-order linearization.

4.3 Structural Assumptions

Block ranks. Known block ranks constrain the candidates examined by OJ, set the per-block support-guessing budgets in PRR, and determine the products of binomial coefficients in blockwise MM.

Support overlap. BAG-Loong's intersection dimension g determines the joint rank $t = r_1 + r_2 - g$. PRR uses the intersection in its admissible subspace dimensions. In MM, the corresponding unknown count after guessing a_1, a_2 coordinates is

$$U = \sum_{i=r_1-g}^{r_1} \binom{d_1 - a_1}{i} \binom{d_2 - a_2}{t-i}.$$

The implementation uses $E = m\binom{d_2-2}{t}$ and the same $q^{a_1 r_1 + a_2 r_2} E U^{\omega-1}$ cost form. It first evaluates the zero-guess candidate and, when needed, searches the two guessing depths.

Ideal blocks. BAG-Piglet, BRA, BRQC, and C-Multi-UR-AG use their ideal-code structure to define block lengths and code dimensions. The assessment does not add a separate multiple-target discount for the ideal-code automorphism orbit.

Multiple syndromes. C-Multi-UR-AG supplies N syndromes sharing a support. Following its specification, the evaluated RSL variants approximate the combinatorial cost by

$$T_{\mathrm{RSL}} \approx T_{\mathrm{RSD}} / 2^N.$$

This subtracts N_1 or N_2 bits from the AGHT and OJ costs for key or message recovery, respectively; MM and SM keep their single-instance costs. For C-Multi-UR-AG-128, the AGHT costs change from 271.8 to 260.8 bits for $N_1 = 11$, and from 268.6 to 252.6 bits for $N_2 = 16$.

4.4 Zero-direction maximal minors (ZD-MM)

Ordinary MM guesses that specified coordinates vanish until the linear system on the maximal minors has enough equations. ZD-MM leaves one zero direction to find inside a block. It computes a kernel basis once, then tests every candidate direction against that basis.

Let $q_0 = 2^d$ be the base field size of the chosen representation, $u_i = n_i - a_i - p_i$ the retained length of block i, and $K' = K - \sum_i a_i$ the dimension after the outer zero guesses. Let V count the admissible minor coordinates, and E the base-field equations in the structured MM system. For direct-sum supports of ranks r_i, $V = \prod_{i:u_i>0} \binom{u_i}{r_i}$. We recompute E after each shortening. In the blockwise equation count, assign $K'+1$ systematic coordinates across the active blocks in a given block order, filling each block in turn. If s_i coordinates fall in block i, its contribution to the equation count is represented by

$$B_i(z) = \sum_{v=\max(0,r_i-s_i)}^{u_i-s_i} \binom{u_i - s_i}{v} z^v.$$

The count is the extension degree times the smallest coefficient $[z^{\sum_i r_i}] \prod_i B_i(z)$ over the block orders, where the sum and product use active blocks. A single block gives $E = m\binom{u_1-K'-1}{r_1}$. The two-block image count used for overlapping supports is given below.

Ordinary MM uses $E \geq V - 1$. ZD-MM starts with $E < V - 1$, giving the model's kernel dimension $\delta = V - E > 1$. Choose $L = r_j + 1$ retained coordinates of an active block j. Their error coefficients span at most r_j dimensions, so some nonzero linear combination vanishes. Enumerate the $Q = (q_0^L - 1)/(q_0 - 1)$ projective directions in those coordinates. Write $x = By$ for the minor vector in the mother kernel. For each direction λ, solve $(\iota_\lambda B)y = 0$, where ι_λ contracts the minor coordinates in block j. Determine the support from a surviving candidate and check it against the public equations.

Let V' and E' be the counts after imposing this extra zero ($u_j \leftarrow u_j - 1,\ K' \leftarrow K' - 1$). We require $\min(E, E') \geq V' - 1$. Set $F = V - V'$ and let G count all output coordinates of the contraction. For direct-sum blocks these are

$$F = \binom{u_j - 1}{r_j - 1} \prod_{i \neq j} \binom{u_i}{r_i}, \qquad G = \binom{u_j}{r_j - 1} \prod_{i \neq j} \binom{u_i}{r_i},$$

with the products again over active blocks. For overlapping supports we sum the same counts over the admissible block degrees, lowering the degree of block j by one in the output.

Let P be the outer-event probability. For a block of length n_i sampled at full rank r_i over the working field, define $\Phi_q(n, r) = \prod_{t=0}^{r-1}(1 - q^{t-n})$. An active block contributes $q_0^{-a_i r_i}\Phi_{q_0}(u_i, r_i)/\Phi_{q_0}(n_i, r_i)$ to P. When the entire block is removed, the numerator uses $n_i - a_i$, the length available to support its original rank. The product also includes any rank event required of a selected row or column. For a packed active block originally sampled over $\mathbb{F}_2$, the denominator is $\Phi_2(dn_i, r_i)$, with n_i here the original physical length in packed units. If this packed block is dropped, its numerator rank factor is $\Phi_2(d(n_i - a_i), r_i)$. When a public message tail is removed first, the rank-event denominator still refers to the original sampled length. Enumerating all Q directions covers the last zero, so Q enters the work per trial.

Let C_{set} denote setup and $C_{\text{rec}}^{\text{MM}}$ and $C_{\text{rec}}^{\text{ZD}}$ denote recovery and verification per candidate for ordinary MM and ZD-MM, respectively. Their expected binary costs are

$$T_{\text{MM}} = \frac{d^2}{P}(C_{\text{set}} + EV^{\omega-1} + C_{\text{rec}}^{\text{MM}}),$$
$$T_{\text{ZD}} = \frac{d^2}{P}(C_{\text{set}} + VE^{\omega-1} + dLF\delta + QG\delta + QG\delta^{\omega-1} + QC_{\text{rec}}^{\text{ZD}}).$$

The wide mother matrix costs $VE^{\omega-1}$ to eliminate. The next three terms prepare the coordinate contractions, update their linear combination in Gray order, and solve the G-by-δ systems. The

factor d in preparation accounts for prescaling by the binary basis elements in packed Piglet models. The outside factor d^2 converts field operations to the common binary cost model. Each method uses the P, E, V of its own configuration. Writing $S = \sum_i n_i$, $N = \sum_i u_i$ and R for the retained joint rank, the estimator charges $C_{\text{set}} = m^2S^3$ and $C_{\text{check}} = m^2(SR^2 + R^3 + SR + S^2) + mS^2$. Recovery adds $RNV + C_{\text{check}}$ for ordinary MM or $RN\delta + C_{\text{check}}$ per direction for ZD-MM, plus any sample completion and decoding required by the selected public relation.

BAG-Loong. For overlap g, use the joint degree $R = r_1 + r_2 - g$ and $V = \sum_{s=r_1-g}^{r_1} \binom{u_1}{s}\binom{u_2}{R-s}$. Set $h = u_1 + u_2 - K' - 1$, $x_i = \max(0, h - u_{3-i})$ and $c = h - x_1 - x_2$. The two-block image gives $E = m\sum_{i=0}^{r_1}\sum_{j=0}^{r_2}\binom{x_1}{i}\binom{x_2}{j}\binom{c}{R-i-j}$, with out-of-range binomial coefficients zero. For the 128-bit key instance the known support element 1 supplies an appended coordinate, giving lengths $(42, 43)$, $K = 42$, $m = 47$, block ranks $(5, 5)$ and $g = 4$. Guessing $(7, 0)$ zeros and puncturing none gives $u = (35, 43)$ and $E = 47[\binom{42}{6} - \binom{7}{6}] = 246\,551\,613$. Here $V = 249\,131\,981$, so $\delta = 2\,580\,368$. In block 2 the 63 directions give $F = 18\,907\,147$, $G = 20\,148\,492$. $-\log_2 P \simeq 35$ gives $\log_2 T_{\text{ZD}} = 113.074$ at $\omega = 2.8$. For messages, a complete ciphertext row in the zero tail of the augmented Gabidulin code removes the message term and yields lengths $(n_0, n_0 + n_1)$ at $K = n_0$.

BAG-Piglet. The two public key equations leave one free ring element among three direct-sum blocks, giving $K = n_0$ in the binary key reduction. At level 256, packing over $\mathbb{F}_4$ gives three blocks of length 39 and rank 6, extension degree 79 and $K = 39$. With $a = (6, 0, 0)$ and no punctures, the mother has $V = \binom{33}{6}\binom{39}{6}^2$. Contraction in the second block uses $Q = (4^7 - 1)/3 = 5\,461$ and yields 254.033 bits. The message estimates compare a four-block ciphertext relation with a three-block first-component relation, including the public zero tail where applicable. The selected representation uses $d = 3$ for the 256-bit message and $d = 2$ for both 384-bit objectives.

BRA. Transposing the binary coordinate matrices of the ideal blocks interchanges the block length and extension degree while preserving each block rank. For BRA-128 key recovery, the original pair $(n_0, m) = (121, 67)$ therefore gives two length-67 blocks over an extension of degree 121, with $K = 67$ and ranks $(4, 5)$. Taking $a = (3, 3)$, $p = (0, 0)$ and contracting block 2 gives $V = \binom{64}{4}\binom{64}{5}$, $\delta = 365\,722\,032\,272$, $Q = 63$ and 112.088 bits. The same two-block search in the original direction gives 190.768 bits. Eliminating the message after transposition removes the entire second ciphertext component, leaving the first-component relation. At level 512, transposing this same message relation lowers the ZD-MM exponent from 886.165 to 522.091.

BRQC. For the three-block projection, the code dimension includes the encoded message dimension μ. If e is the noise block, the remaining message kernel has dimension $\kappa = \max(0, \mu - a_e)$ in the searched generic projection. Count equations with the image length $u_e - \kappa$ and dimension $K' - \kappa$, and minor variables with the physical length u_e. The 128-bit message instance uses $m = 127$, $K = 122$, lengths $(119, 119, 119)$, ranks $(5, 5, 8)$, $a = (0, 0, 0)$ and $p = (0, 0, 16)$. Thus $\kappa = 3$, the physical noise length is 103 and its image length is 100. The formal equation count gives 185.575 bits. The discussion after Table 7 quantifies the effect of the known common spectral components on the message estimates.

C-Multi-UR-AG. The key instance uses one sample and the known support element 1, giving $(n, K, r) = (2n_0 + 1, n_0, w_1)$. A column with full message image gives the projected message instance $(2n_0 + N_1, n_0 + k_{\text{msg}}, w_2)$. For the 128-bit key, $(n, K, r, m) = (71, 35, 8, 79)$. With 13 outer zeros and no punctures, $V = \binom{58}{8}$, $E = 79\binom{35}{8}$, and $Q = 2^9 - 1 = 511$ give 168.212 bits. The corresponding message dimensions $(81, 38, 9, 79)$ and 15 outer zeros give 209.247 bits. Table 7 reports both objectives at every level.

4.5 Results

Table 6 gives the lowest cost for each recovery objective. The first pair of columns covers the attacks in each specification. The second also includes ordinary MM with the structural reductions and ZD-MM. The declared value is the submission's lowest classical estimate across all attacks and both objectives.

Table 6: Minimum classical decoding costs $\log_2 T$ by recovery objective and the minimum declared by each submission.

	Original attacks		All evaluated attacks		
Parameter set	Key	Message	Key	Message	Declared
BAG-Loong-128	122.2	**120.5**	**113.1**	115.7	139.0
BAG-Loong-256	**221.5**	234.7	**212.3**	229.0	271.0
BAG-Loong-384	**310.3**	349.1	**299.1**	342.0	403.0
BAG-Loong-512	450.8	**393.5**	444.1	**386.7**	534.0
BAG-Piglet-128	**148.1**	161.3	**114.6**	156.7	148.0
BAG-Piglet-256	**268.8**	297.5	**254.0**	283.0	269.0
BAG-Piglet-384	**397.1**	442.3	**394.2**	442.3	397.0
BAG-Piglet-512	**552.4**	563.2	**552.4**	563.2	552.0
BRA-128	194.6	**189.1**	**112.1**	160.3	189.0
BRA-256	**304.7**	321.8	**161.8**	264.3	304.0
BRA-512	**556.0**	567.1	**316.8**	522.1	556.0
BRQC-128	216.8	**181.7**	211.9	**181.7**	181.0
BRQC-256	309.2	**295.4**	304.5	**295.4**	295.0
BRQC-512	584.6	**566.8**	578.9	**566.8**	566.0
C-Multi-UR-AG-128	172.8	**170.9**	**168.2**	170.9	170.0
C-Multi-UR-AG-256	296.3	**276.9**	283.9	**276.9**	276.0
C-Multi-UR-AG-512	**531.2**	580.5	**515.8**	580.5	530.0

For BAG-Loong, the original attacks give minima of 120.5, 221.5, 310.3, and 393.5 bits at targets of 128, 256, 384, and 512 bits. Adding ZD-MM lowers the four minima to 113.1, 212.3, 299.1, and 386.7 bits, respectively.

Ordinary MM and ZD-MM

Table 7 compares the declared MM cost with our estimates for ordinary MM and ZD-MM. Both methods use the same structural reductions and cost model. Each selects its own guesses and punctures, so their cost difference measures the gain from the zero-direction search. For BRA and BRQC, the declared key and message costs correspond to the $2n$ and $3n$ instances, respectively. For C-Multi-UR-AG they correspond to w_1 and w_2. The public message projection in Section 4.4 changes the code dimension for w_2 from n_0 to $n_0 + k_{\mathrm{msg}}$.

Table 7: Classical MM work factors by recovery objective ($\log_2 T$). Declared costs come from Table 5 of each BAG specification (PDF pages 22 and 21) and Table 6 of the BRA, BRQC and C-Multi-UR-AG specifications (PDF pages 30, 29 and 28, respectively) [1].

	Key recovery			Message recovery		
Parameter set	Declared	Ordinary MM	ZD-MM	Declared	Ordinary MM	ZD-MM
BAG-Loong-128	158	117.872	113.074	139	120.520	115.749
BAG-Loong-256	286	217.134	212.315	292	234.656	228.984
BAG-Loong-384	403	304.862	299.061	469	348.685	342.041
BAG-Loong-512	626	450.783	444.080	534	393.517	386.708
BAG-Piglet-128	148	118.416	114.644	167	156.728	158.899
BAG-Piglet-256	269	261.798	254.033	298	283.629	283.041
BAG-Piglet-384	397	402.373	394.160	442	458.279	450.550
BAG-Piglet-512	552	574.937	567.599	563	607.713	603.808
BRA-128	194	116.025	112.088	189	164.044	160.318
BRA-256	304	166.503	161.840	321	268.954	264.251
BRA-512	556	321.752	316.802	567	526.495	522.091
BRQC-128	216	216.637	211.948	181	185.821	185.575
BRQC-256	309	309.232	304.468	295	334.570	328.490
BRQC-512	584	584.660	578.878	566	598.130	590.891
C-Multi-UR-AG-128	172	172.821	168.212	170	213.369	209.247
C-Multi-UR-AG-256	296	287.089	283.854	276	319.183	315.661
C-Multi-UR-AG-512	530	519.847	515.782	580	662.313	657.968

ZD-MM costs less than ordinary MM for 33 of the 34 objectives. The largest reduction is 8.213 bits for BAG-Piglet-384 key recovery. For BAG-Piglet-128 message recovery, ordinary MM costs 156.728 bits and enumerating the local directions raises the total to 158.899. ZD-MM falls below the declared MM cost for the same objective in 25 of 34 comparisons. Taking the cheaper of key and message recovery for each parameter set, ZD-MM falls below the lowest declared cost across all attacks for 12 of 17 sets and below the nominal target for 9 of 17. The declared minimum of 271 for BAG-Loong-256 comes from OJ, while the declared key and message MM exponents are 286 and 292. Across all attacks in Table 6, ZD-MM gives the lowest cost for 24 objectives, structured ordinary MM for one, and the original attacks for nine.

BRQC message recovery. The ZD-MM message estimates of 185.575, 328.490 and 590.891 bits use the ordinary MM equation count. The 128-bit three-block configuration uses puncturing alone. The ring multipliers act diagonally in frequency coordinates. The three blocks can share a frequency at which the public check supplies only two output directions. Puncturing can preserve this structural dependency. Requiring a block of zero leading polynomial coefficients gives a shortened evaluation space that is injective after deletion of the overlapping frequencies (the MDS prefix). With this prefix, the paired ordinary MM and ZD-MM message exponents are 216.637/211.948, 338.373/332.793 and 598.137/590.891. The prefix gives the same three ZD-MM key exponents to three decimal places. Including the cost of constructing the MM coefficients gives ZD-MM message exponents of 228.669, 352.961 and 611.804, respectively.

Tables 14 and 15 give the guesses, punctures and stage costs for all 34 ZD-MM estimates. The binary three- and four-block BAG-Piglet-128 message searches cover 608,321 retained-length profiles after symmetry reduction. The 256-bit $d = 2, 3$ and 384-bit $d = 2$ packed models also use exhaustive ZD-MM searches for both objectives. For the remaining multiblock models we compare feasible configurations and refine them by local search.

5 Conclusion

We compare the estimated classical work factors, $\log_2 T$, with the scheme specifications and stated security targets. Comparisons refer to individual parameter sets and recovery routes, or to explicitly named attack algorithms. Estimates within 3 bits of a declared value are considered close. All conditional combined-ciphertext (CT) estimates assume access to the full masking term at no recovery cost (Table 2).

5.1 Lower estimates below the target

ZD-MM gives work factors below both the declared minimum and the security target for all four BAG-Loong sets, BAG-Piglet-128 and -256, and all three BRA sets (Table 8). Message recovery gives the lowest cost for BAG-Loong-512. Key recovery gives the lowest cost for the other eight sets. NSS-HQC key recovery also falls below the declared value and target at levels 256, 384 and 512 [10].

Table 8: Classical decoding costs below the declared values and stated targets, in bits. For the rank-metric schemes, the declared column gives the specification's minimum across attacks and recovery routes; the route column identifies the evaluated minimum.

Parameter set	Route	Declared	Estimate	Target
BAG-Loong-128	Key	139.0	113.1	128.0
BAG-Loong-256	Key	271.0	212.3	256.0
BAG-Loong-384	Key	403.0	299.1	384.0
BAG-Loong-512	Message	534.0	386.7	512.0
BAG-Piglet-128	Key	148.0	114.6	128.0
BAG-Piglet-256	Key	269.0	254.0	256.0
BRA-128	Key	189.0	112.1	128.0
BRA-256	Key	304.0	161.8	256.0
BRA-512	Key	556.0	316.8	512.0
NSS-HQC-256	Key	256.0	247.6	256.0
NSS-HQC-384	Key	384.0	364.2	384.0
NSS-HQC-512	Key	512.0	480.6	512.0

For NSS-HQC, baseline message recovery is cheaper than key recovery at these three levels, with costs of 226.0, 332.5, and 440.9 bits, respectively (Table 3). Thus the baseline estimates already fall below the security targets. The conditional CT estimates also fall below both the declared lower bounds and the targets.

The specification's weight-loss extrapolation understates the reduction in message-recovery cost. Across all four parameter levels, our Stern estimates decrease by 13.6–39.7 bits from key recovery to baseline message recovery. These reductions exceed the respective claimed losses of less than 3, 5, 7, and 9 bits.

5.2 Lower estimates that remain above the target

Table 9 lists recovery estimates more than 3 bits below the declared values that remain above the security target.

Table 9: Lower platform estimates that remain above the stated target, in bits.

Parameter set	Route	Declared	Estimate	Target
C-Multi-UR-AG-128	Message	256.0	252.6	128.0
C-Multi-UR-AG-512	Key (ZD-MM)	530.0	515.8	512.0
NSS-HQC-128	Key	188.2	159.8	128.0

For C-Multi-UR-AG-128, 252.6 bits is the RSL-AGHT cost for message recovery. The lowest evaluated message cost is 170.9 bits with MM, compared with the declared MM cost of 170 bits (Table 6).

For NSS-HQC-128, the 146.2-bit baseline message estimate and the 147.0-bit conditional CT estimate remain above the 128-bit target. The CT estimate is, however, below the declared 191-bit lower bound.

For Mito, the single-target CT estimates without block-weight conditioning reproduce the specification's values within 0.05 bits. Accounting for multiple targets and known block weights, and optimizing over the evaluated algorithms, lowers the costs by 13.1–62.4 bits across all nine parameter sets [12]. These CT estimates remain above the security targets.

5.3 Estimates close to the specifications

Table 10 compares the estimated and declared minimum costs for the indicated schemes and attack routes. Each pair gives the specification value followed by the evaluated estimate, in bits. For BAG-Piglet, the four minima over the original attacks differ from the declared costs by at most 0.4 bits, although most PRR estimates exceed their declared values.

Table 10: Declared and evaluated minimum classical costs that differ by at most 3 bits. The rank-metric rows use the original attacks. A dash means that the parameter set is not present.

Scheme / route	128	256	384	512
BAG-Piglet	148.0/148.1	269.0/268.8	397.0/397.1	552.0/552.4
BRA	189.0/189.1	304.0/304.7	—	556.0/556.0
BRQC	181.0/181.7	295.0/295.4	—	566.0/566.8
C-Multi-UR-AG	170.0/170.9	276.0/276.9	—	530.0/531.2
HARE, Key	169.0/170.6	273.0/275.1	397.0/399.3	529.0/531.1
TriQ-KEM, Key	147.0/146.6	277.1/277.0	404.6/404.6	532.4/532.5
TriQ-KEM, CT	146.7/145.9	277.0/277.0	404.1/404.1	532.1/532.1
TRIKE, Key	133.8/133.8	266.3/266.3	398.3/398.2	530.9/530.9
TRIKE, Message	131.7/131.3	264.5/264.4	398.1/398.1	525.6/525.6
Mito-1, Key	185.8/185.1	283.0/282.4	—	541.1/540.4
Mito-1-E, Key	185.6/185.0	282.8/282.3	—	540.9/540.3
Mito-2-E, Key	200.2/199.7	298.5/297.3	—	558.7/557.7
QCTM, Message	150.9/150.9	272.9/272.9	—	530.3/530.3

Across all five rank-metric schemes, 110 of the 132 estimates for the original attacks are within one bit of the specification values (Section 4, Table 6).

The HARE and TRIKE v1.6 columns in Table 10 show sets 2/5/7/9 and 1/5/7/9, respectively. The omitted HARE-1/3 key estimates and TRIKE-3 key and message estimates also satisfy the 3-bit criterion [7, 8]. For HARE and TriQ-KEM, baseline message costs equal key costs at every parameter set [11].

For all nine Mito parameter sets, neither message route nor the parity-constrained key model lowers the baseline key-recovery cost.[1] The QCTM estimates assume no memory limit [14].

[1]For Mito-1-E-256, we use the ring length $r = 32261$. The specification's cost table incorrectly uses the encoded length $n' = 32256$ instead [12].

5.4 Higher estimates under different cost models

Table 11 compares the higher estimates with the specification values. HARE's specification incorporates parity constraints and ciphertext truncation into its conservative CT analysis. Our CT model uses the full-length masking term (Table 2) and gives higher costs for all six parameter sets [8].

For QUBE, the specification subtracts the full logarithmic reduction in support-space size from the time exponent. Our model instead computes the gain through the decoding success probability under the known block weights [13]. The resulting key and CT estimates are higher for all five parameter sets, with key recovery remaining the cheapest evaluated route.

For UVW-KEM, the specification uses an idealized q-ary ISD formula, whereas our estimates use finite-parameter decoding cost models [15]. The specification also counts the scalar-normalization gain twice, thereby underestimating the work factor. Our message-recovery estimates are higher for all three parameter sets.

Table 11: Higher classical estimates under different cost models, in bits. Each pair gives the specification value followed by the platform estimate.

Scheme / route	128	256	384	512
HARE, CT	161.0/171.8	263.0/274.4	387.0/399.7	516.0/528.9
QUBE, Key	136.5/145.1	263.7/273.5	392.8/402.5	520.3/530.4
UVW-KEM, Message	128.4/146.5	256.8/275.9	—	513.2/533.3

5.5 Comparisons with security targets

Nine of the 17 rank-metric parameter sets fall below target with ZD-MM (Table 8). The lowest estimates for BAG-Piglet-384 and -512, all three BRQC sets and all three C-Multi-UR-AG sets remain above target. Among the Hamming-metric schemes, the below-target classical estimates concern NSS-HQC at levels 256, 384 and 512.

BIKE-MLThre and HEP-QC specify security targets rather than per-algorithm ISD work factors [6, 9]. Their estimated classical and quantum decoding costs exceed the corresponding stated targets as in Table 12.

Table 12: Security targets and minimum decoding cost estimates, in bits. Minima are taken over the evaluated recovery routes.

	Classical		Quantum	
Parameter set	Target	Estimate	Target	Estimate
BIKE-MLThre-128	128	145.9	80	95.2
BIKE-MLThre-256	256	275.9	128	162.4
BIKE-MLThre-512	512	536.1	256	297.8
HEP-QC-1	128	144.8	64	98.6
HEP-QC-3	192	213.1	96	134.1
HEP-QC-5	256	275.3	128	165.9
HEP-QC-7	512	535.8	256	298.1

Among the evaluated Hamming quantum attacks, only NSS-HQC-512 message recovery falls below its stated target: the baseline and conditional CT estimates are 251.6 and 251.2 bits, respectively, against a 256-bit target (Table 3).

A ZD-MM operation counts

Table 13: Working parameters for ZD-MM key and message recovery.

Set	Objective	(d, m, K)	Block lengths (n_i)	Block ranks (r_i)	j	g or (μ, i_e)
BAG-Loong						
128	Key	$(1, 47, 42)$	$(42, 43)$	5×2	2	$g = 4$
	Message	$(1, 47, 42)$	$(42, 52)$	5×2	1	$g = 3$
256	Key	$(1, 67, 65)$	$(65, 66)$	$(5, 6)$	1	$g = 3$
	Message	$(1, 67, 65)$	$(65, 77)$	6×2	1	$g = 3$
384	Key	$(1, 83, 83)$	$(83, 84)$	6×2	2	$g = 3$
	Message	$(1, 83, 83)$	$(83, 97)$	7×2	1	$g = 4$
512	Key	$(1, 97, 104)$	$(104, 105)$	7×2	1	$g = 4$
	Message	$(1, 97, 104)$	$(104, 119)$	7×2	2	$g = 4$
BAG-Piglet						
128	Key	$(1, 47, 43)$	43×3	4×3	1	—
	Message	$(1, 47, 86)$	43×3	$(2, 3, 3)$	3	—
256	Key	$(2, 79, 39)$	39×3	6×3	2	—
	Message	$(3, 79, 52)$	26×4	$(3, 3, 4, 4)$	2	—
384	Key	$(2, 103, 53)$	53×3	7×3	2	—
	Message	$(2, 103, 106)$	53×4	4×4	2	—
512	Key	$(1, 131, 125)$	125×3	8×3	3	—
	Message	$(1, 131, 250)$	$(125, 125, 120, 125)$	$(4, 4, 5, 5)$	1	—
BRA						
128	Key	$(1, 121, 67)$	67×2	$(4, 5)$	2	—
	Message	$(1, 121, 67)$	67×2	5×2	2	—
256	Key	$(1, 161, 83)$	83×2	5×2	1	—
	Message	$(1, 161, 83)$	83×2	6×2	1	—
512	Key	$(1, 221, 127)$	127×2	6×2	2	—
	Message	$(1, 221, 127)$	127×2	$(7, 8)$	1	—
BRQC						
128	Key	$(1, 127, 119)$	119×2	5×2	1	—
	Message	$(1, 127, 122)$	119×3	$(5, 5, 8)$	1	$\mu = 3,\ i_e = 3$
256	Key	$(1, 163, 161)$	161×2	$(5, 6)$	1	—
	Message	$(1, 163, 164)$	161×3	$(6, 7, 8)$	3	$\mu = 3,\ i_e = 3$
512	Key	$(1, 229, 227)$	227×2	$(6, 7)$	1	—
	Message	$(1, 229, 230)$	227×3	8×3	1	$\mu = 3,\ i_e = 2$
C-Multi-UR-AG						
128	Key	$(1, 79, 35)$	71	8	1	—
	Message	$(1, 79, 38)$	81	9	1	—
256	Key	$(1, 127, 45)$	91	10	1	—
	Message	$(1, 127, 48)$	105	11	1	—
512	Key	$(1, 181, 67)$	135	12	1	—
	Message	$(1, 181, 71)$	153	14	1	—

The base field is $\mathbb{F}_{2^d}$ and the extension degree over that field is m. K and the block lengths are measured after the public reduction, before the outer guesses a and punctures p in Tables 14 and 15. The notation $x \times \ell$ denotes ℓ consecutive blocks with value x. j numbers the contraction block starting at 1, including any preceding blocks subsequently removed. g is the support intersection dimension. In BRQC message rows, μ is the public message dimension and i_e locates its error block in the listed order.

Table 14: ZD-MM parameters and expected work by stage for key recovery.

	Block coordinates			Expected cost of each stage						
Set	a	p	$-\log_2 P$	Setup	Kernel	Prep.	Update	Solve	Recover	Total
BAG-Loong	$\omega = 2.8$									
128	$(7, 0)$	$(0, 0)$	35.000	65.337	113.072	83.057	86.541	103.580	73.132	**113.074**
256	$(21, 0)$	$(0, 0)$	105.000	138.232	212.292	175.919	179.400	206.367	154.470	**212.315**
384	$(29, 0)$	$(0, 0)$	174.000	208.901	299.011	257.696	261.951	294.173	231.545	**299.061**
512	$(43, 0)$	$(1, 0)$	301.000	337.322	443.903	398.272	403.353	440.969	366.703	**444.080**
BAG-Piglet	$\omega = 2.8$									
128	$(0, 0, 5)$	$(0, 0, 38)$	20.000	52.143	114.635	81.469	84.206	107.203	63.128	**114.644**
256	$(6, 0, 0)$	$(0, 0, 0)$	72.000	107.219	251.243	197.650	206.456	253.807	156.569	**254.033**
384	$(12, 0, 0)$	$(0, 0, 0)$	168.000	205.312	390.594	324.092	334.680	394.033	270.198	**394.160**
512	$(0, 0, 30)$	$(0, 0, 1)$	240.000	279.719	567.116	469.474	475.413	565.788	374.977	**567.599**
BRA	$\omega = 2$									
128	$(3, 3)$	$(0, 0)$	27.000	62.036	111.166	106.458	109.944	109.944	81.559	**112.088**
256	$(0, 13)$	$(0, 0)$	65.000	101.787	161.611	154.548	158.011	158.011	124.252	**161.840**
512	$(0, 32)$	$(0, 0)$	192.000	231.542	315.849	310.458	314.717	314.717	268.041	**316.802**
BRQC	$\omega = 2$									
128	$(0, 21)$	$(0, 0)$	105.000	142.662	211.918	201.849	205.290	205.290	167.436	**211.948**
256	$(36, 0)$	$(0, 0)$	180.000	219.690	304.292	296.843	300.282	300.282	254.335	**304.468**
512	$(71, 0)$	$(0, 0)$	426.000	468.158	578.863	567.074	571.303	571.303	511.806	**578.878**
C-Multi-UR-AG	$\omega = 2$									
128	(13)	(0)	104.000	135.057	165.628	160.924	166.937	166.937	147.766	**168.212**
256	(20)	(0)	200.000	233.501	277.308	275.100	282.835	282.835	256.191	**283.854**
512	(34)	(0)	408.000	444.230	508.030	505.309	514.775	514.775	477.842	**515.782**

a counts the coordinates actually guessed zero before the local direction is supplied, and p counts punctures in the same block order. P is the probability of the outer events. Each stage is reported as $c_i = \log_2(C_i/P)$, where C_i is its single-batch work in binary-field operations. Setup covers the initial preprocessing. Kernel computes the kernel of the underdetermined MM system. Prep. constructs the coordinate contractions. Update and Solve charge all local directions. Recover includes the applicable completion, decoding, and verification work. The total is $\log_2(\sum_i 2^{c_i})$ using the unrounded exponents.

Table 15: ZD-MM parameters and expected work by stage for message recovery.

	Block coordinates			Expected cost of each stage						
Set	a	p	$-\log_2 P$	Setup	Kernel	Prep.	Update	Solve	Recover	Total
BAG-Loong	$\omega = 2.8$									
128	$(5, 0)$	$(0, 0)$	25.000	55.773	115.663	85.182	88.674	111.634	69.003	**115.749**
256	$(18, 0)$	$(0, 0)$	108.000	141.581	228.795	190.026	194.310	225.960	164.675	**228.984**
384	$(29, 0)$	$(1, 0)$	203.000	238.226	341.827	297.586	302.677	339.179	267.173	**342.041**
512	$(34, 0)$	$(0, 0)$	238.000	274.603	386.624	338.517	343.573	382.577	305.634	**386.708**
BAG-Piglet	$\omega = 2.8$									
128	$(41, 0, 0)$	$(2, 2, 0)$	83.415	115.558	158.898	129.327	131.302	147.928	117.158	**158.899**
256	$(17, 0, 0, 0)$	$(1, 0, 0, 0)$	153.000	188.879	281.540	243.284	249.007	282.413	217.353	**283.041**
384	$(37, 0, 0, 0)$	$(16, 0, 0, 0)$	296.000	334.557	450.444	401.790	406.966	446.740	367.029	**450.550**
512	$(0, 96, 0, 0)$	$(0, 29, 0, 0)$	384.000	424.921	603.807	532.587	535.254	593.423	474.004	**603.808**
BRA	$\omega = 2$									
128	$(0, 14)$	$(0, 0)$	70.000	105.036	159.189	154.869	158.374	158.374	127.240	**160.318**
256	$(0, 26)$	$(0, 0)$	156.000	192.787	263.127	258.058	262.329	262.329	223.139	**264.251**
512	$(54, 0)$	$(0, 0)$	378.000	417.542	519.662	515.656	520.774	520.774	464.696	**522.091**
BRQC	$\omega = 2$									
128	$(0, 0, 0)$	$(0, 0, 16)$	0.000	39.417	185.309	178.498	181.940	181.940	106.367	**185.575**
256	$(16, 0, 0)$	$(0, 0, 0)$	96.000	137.445	326.644	321.116	327.007	327.007	229.178	**328.490**
512	$(0, 39, 0)$	$(0, 0, 0)$	312.000	355.913	590.355	582.319	588.191	588.191	467.698	**590.891**
C-Multi-UR-AG	$\omega = 2$									
128	(15)	(0)	135.000	166.627	205.143	201.291	208.154	208.154	184.949	**209.247**
256	(20)	(0)	220.000	254.120	308.383	306.054	314.649	314.649	283.009	**315.661**
512	(38)	(0)	532.000	568.772	648.500	645.700	656.966	656.966	612.179	**657.968**

a counts the coordinates actually guessed zero before the local direction is supplied, and p counts punctures in the same block order. P is the probability of the outer events. Each stage is reported as $c_i = \log_2(C_i/P)$, where C_i is its single-batch work in binary-field operations. Setup covers the initial preprocessing. Kernel computes the kernel of the underdetermined MM system. Prep. constructs the coordinate contractions. Update and Solve charge all local directions. Recover includes the applicable completion, decoding, and verification work. The total is $\log_2(\sum_i 2^{c_i})$ using the unrounded exponents.

For BRQC these rows use the equation-count model. At level 128 the selected three-block instance has no externally guessed zeros and punctures 16 coordinates in the last block.

References

[1] Institute of Commercial Cryptography Standards. Announcement of the first-round public-key cryptographic algorithm candidates in the global call for next-generation commercial cryptographic algorithms, September 2026. URL https://www.niccs.org.cn/symmbzyjy/tzgg/pc/content/2100859557292625920/content_2100859557292625920.html. 20 September 2026. Candidate list: https://www.niccs.org.cn/niccs/Round1Additional/pc/list.html.

[2] Andre Esser, Javier A. Verbel, Floyd Zweydinger, and Emanuele Bellini. SoK: Cryptographic Estimators - a software library for cryptographic hardness estimation. In *AsiaCCS*. ACM, 2024. doi: 10.1145/3634737.3645007.

[3] Kévin Carrier, Valérian Hatey, Laura Luzzi, and Jean-Pierre Tillich. Multilevel amortized gaussian elimination in information-set decoding: Applications to HQC and PCG. Cryptology ePrint Archive, Paper 2026/1498, 2026. URL https://eprint.iacr.org/2026/1498.

[4] Ghazal Kachigar and Jean-Pierre Tillich. Quantum information set decoding algorithms. In Tanja Lange and Tsuyoshi Takagi, editors, *PQCrypto 2017*, pages 69–89. Springer, Cham, 2017. doi: 10.1007/978-3-319-59879-6_5.

[5] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. North Holland, 1977. ISBN 978-0444850102.

[6] Yongbin Liu, Zhenxing Niu, Haichang Gao, Yulong Shen, Xutong Mu, and Ke Cheng. BIKE_MLThre: A machine-learning-assisted threshold-selection variant of BIKE. Algorithm specification, submission package, 2026. URL https://www.niccs.org.cn/niccs/Round1Additional/pc/content/content_2101560843495362560.html.

[7] Tianrui Wang, Yuxuan Li, Jing Tian, Yangcan Zhou, Zijian Zhang, and Anyu Wang. TRIKE: Algorithm specifications and supporting documentation. Algorithm specification, ICCS version, 2026. URL https://pqc-trike.info/downloads/trike-specification-iccs-v1.6.pdf. Version 1.6.

[8] Tianrui Wang, Qicheng Teng, Jipeng Li, Anyu Wang, Jun Zhang, Bo Pang, Chaojian Shen, Zhiyuan Qiu, Chunhuan Zhao, and Sihuang Hu. HARE: Algorithm specifications and supporting documentation. Algorithm specification, submission package, 2026. URL https://www.niccs.org.cn/niccs/Round1Additional/pc/content/content_2101560844942397440.html. Version 1.5.

[9] Chengju Li, Tianrui Liu, Chunyu Gan, Zhongyu Huang, Shuying Dong, and Xinran Wang. Hybrid equivalent punctured and quasi-cyclic (HEP-QC). Algorithm specification, East China Normal University submission package, June 2026. URL https://www.niccs.org.cn/niccs/Proposal/Public-Key%20Cryptographic%20Algorithms/Round%201%20candidates/HEP-QC.zip.

[10] Yulong Shen, Xutong Mu, Shuaiping Dong, Zidong Zhou, Ahmat Bahadir, Junbo Ni, Haoyu Zhang, Jinxuan Cui, Ke Cheng, Anxiao Song, Zhenxing Niu, and Lele Zheng. NSS-HQC. Algorithm specification, Xidian University submission package, April 2026. URL https://www.niccs.org.cn/niccs/Round1Additional/pc/content/content_2101560863191814144.html.

[11] Lan Luo, Chaofeng Guan, Yunlei Zhao, Hong Wang, Jianhua Hou, and Tong Yu. TriQ-KEM: Key encapsulation mechanism specification and supporting documentation. Algorithm specification, submission package, 2026. URL https://www.niccs.org.cn/niccs/Round1Additional/pc/content/content_2101560881558671360.html.

[12] Changlun Li, Chengju Li, Liang Bai, Qiping Lin, Tianrui Liu, Yan Ding, Chunyu Gan, and Zhongyu Huang. MITO: Algorithm specification and supporting documentation. Algorithm specification, submission package, June 2026. URL https://www.niccs.org.cn/niccs/Round1Additional/pc/content/content_2101560862776578048.html.

[13] Li-Ping Wang, Yi Tang, Haofei Wang, and Peiying Xu. QUBE: Quasi-cyclic with unbalanced & multi-block errors. Algorithm specification, submission package, 2026. URL https://www.niccs.org.cn/niccs/Round1Additional/pc/content/content_2101560881000828928.html.

[14] Qin Yue, Fengwei Li, Licheng Wang, Huan Sun, Yufeng Song, Yiwei Jiang, and Jingjing Zhang. Quasi-cyclic twisted McEliece key encapsulation mechanism (QCTM): Algorithm specification. Submission version for commercial cryptographic scheme solicitation, June 2026. URL https://www.niccs.org.cn/niccs/Round1Additional/pc/content/content_2101560872431865856.html.

[15] Fangguo Zhang, Yang Yang, Kuojie Xiao, Xiaofeng Chen, Chen Li, and Haibo Tian. UVW-KEM: Algorithm description document. Algorithm specification, submission package, April 2026. URL https://www.niccs.org.cn/niccs/Round1Additional/pc/content/content_2101560881697083392.html.

[16] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962. doi: 10.1109/TIT.1962.1057777.

[17] Pil Joong Lee and Ernest F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In C. G. Günther, editor, *EUROCRYPT'88*, volume 330 of *LNCS*, pages 275–280. Springer, Berlin, Heidelberg, May 1988. doi: 10.1007/3-540-45961-8_25.

[18] Jacques Stern. A method for finding codewords of small weight. In Gérard Cohen and Jacques Wolfmann, editors, *Coding Theory and Applications*, pages 106–113, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg. ISBN 978-3-540-46726-7.

[19] Ilya Dumer. On minimum distance decoding of linear codes. In *Proceedings of the Fifth Joint Soviet-Swedish International Workshop on Information Theory*, pages 50–52, Moscow, 1991.

[20] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: Ball-collision decoding. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 743–760. Springer, Berlin, Heidelberg, August 2011. doi: 10.1007/978-3-642-22792-9_42.

[21] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In *Advances in Cryptology – EUROCRYPT 2012*, pages 520–536. Springer, 2012.

[22] Andre Esser and Floyd Zweydinger. New time-memory trade-offs for subset sum – improving ISD in theory and practice. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, pages 360–390. Springer, 2023.

[23] Andre Esser and Emanuele Bellini. Syndrome decoding estimator. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part I*, volume 13177 of *LNCS*, pages 112–141. Springer, Cham, March 2022. doi: 10.1007/978-3-030-97121-2_5.

[24] Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In *Advances in Cryptology – EUROCRYPT 2015, Part I*, pages 203–228. Springer, 2015.

[25] Leif Both and Alexander May. Decoding linear codes with high error rate and its impact for LPN security. In Tanja Lange and Rainer Steinwandt, editors, *PQCrypto 2018*, pages 25–46. Springer, Cham, 2018. doi: 10.1007/978-3-319-79063-3_2.

[26] Christiane Peters. Information-set decoding for linear codes over F_q. In Nicolas Sendrier, editor, *PQCrypto 2010*, pages 81–94. Springer, Berlin, Heidelberg, May 2010. doi: 10.1007/978-3-642-12929-2_7.

[27] Frédéric Magniez, Ashwin Nayak, Jeremie Roland, and Miklos Santha. Search via quantum walk. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 575–584. ACM Press, June 2007. doi: 10.1145/1250790.1250874.

[28] Gregory V. Bard. Accelerating cryptanalysis with the method of four russians. Cryptology ePrint Archive, Report 2006/251, 2006. URL `https://eprint.iacr.org/2006/251`.

[29] Nicolas Sendrier. Decoding one out of many. In Bo-Yin Yang, editor, *PQCrypto 2011*, pages 51–67. Springer, Berlin, Heidelberg, November / December 2011. doi: 10.1007/978-3-642-25405-5_4.

[30] Andre Esser, Alexander May, and Floyd Zweydinger. McEliece needs a break - solving McEliece-1284 and quasi-cyclic-2918 with modern ISD. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 433–457. Springer, Cham, May / June 2022. doi: 10.1007/978-3-031-07082-2_16.

[31] Yongcheng Song, Rongmao Chen, Fangguo Zhang, Xinyi Huang, Jian Weng, and Huaxiong Wang. (Interleaved) extended gabidulin codes, more analysis on blockwise rank decoding problem, and their applications to cryptosystems. Cryptology ePrint Archive, Report 2025/668, 2025. URL `https://eprint.iacr.org/2025/668`.