# When Confirmation Tags Are Not Enough: Full Key Recovery from Ordinary Decapsulation Outputs in HCR-KEM

Rafik Hamza

Institute for International Strategy, Tokyo International University, Tokyo, Japan

**Abstract**

Reconciliation allows lattice-based key encapsulation mechanisms to derive a common secret despite noise. Yet confirming the resulting verifier word need not establish that the ciphertext itself is valid. This distinction matters when a design seeks to avoid full ciphertext re-encryption by checking only a confirmation tag. We introduce HCR-KEM, a source-available reconciliation-based research construction, to examine the security consequences of that choice.

We recover all 1,536 secret coefficients using only ordinary 32-byte decapsulation outputs, without an explicit validity flag. Source-admitted unit-monomial ciphertexts keep 289 non-target verifier bits fixed, allowing locally predicted-key comparisons to isolate signed relations. A publicly checkable signed-graph certificate then enables complete reconstruction. Building on this structure, graph-aware peeling uses recovered coefficients to reduce the cost of subsequent ordered searches.

In reference C experiments on 50 fresh matched keys, all 150 recoveries succeeded. Graph-aware peeling reduced mean online decapsulation calls from 4,382.54 for the depth-four scalar baseline to 3,565.18, an 18.65% reduction. To distinguish this leakage mechanism from cases where it does not arise, we also formulate a conditional reconciliation-independence criterion and examine the guaranteed-rejection branch of standard ML-KEM. Together, these results show why the security of a reconciliation-based KEM must be assessed from the ciphertexts it admits and the information exposed by its complete decapsulation interface, rather than from verifier-word confirmation alone.

**Keywords:** Chosen-ciphertext attacks, Lattice-based cryptography, Key encapsulation, key-mismatch attacks, Reconciliation

## 1 Introduction

Reconciliation can reduce communication overhead or decoding failures, but these benefits do not establish ciphertext authenticity. The distinction becomes important when a design confirms a recovered verifier word instead of checking the ciphertext that produced it. We introduce HCR-KEM as a source-available research construction to study whether such confirmation can enforce ciphertext consistency. Its construction and cryptanalysis together provide a concrete counterexample to treating word confirmation as complete ciphertext validation.

Full Fujisaki–Okamoto (FO) decapsulation decrypts and re-encrypts a ciphertext, comparing it to the received ciphertext before releasing an accepted key. FIPS 203

Email: rhamza@tiu.ac.jp (Rafik Hamza)

Date of this document: 2026-11-23.

Algorithm 18 uses precisely this check for ML-KEM [19]. Could a reconciliation-based KEM replace that work by authenticating only its verifier word? In this architecture, the sender derives the verifier word only after sampling its ephemeral noise and evaluating the noisy inner product; it is not a message selected in advance to regenerate the ciphertext. The usual message-derived FO re-encryption check therefore cannot be inserted into this particular interface unchanged, though a redesigned encryption-based or actively authenticated construction remains possible [20]. The construction uses two noisy public module products, $t = As + e$ and $\boldsymbol{p} = Ax + f$, and selects a forest of reconciliation positions. Its activity filter requires at least 64 of 128 quadratic CRT factors to be active, while the SHA3-256 tag confirms the derived verifier word. The activity check is intended to exclude trivial all-zero and low-rank cores; the tag checks whether the receiver reconstructs the proposed verifier word. Neither condition imposes a norm bound on the ciphertext core or validates the complete ciphertext. We examine whether these components can provide ciphertext consistency without a full re-encryption check; no bandwidth or failure-rate advantage is assumed. The relevant question is therefore not whether the filter rejects some malformed inputs, but whether an admitted input can still produce a predictable, secret-dependent comparison.

When a modified ciphertext changes the receiver's word $V_r$, its confirmation tag disagrees with the tag computed for the sender's hypothesized $V_s$. The ordinary Decaps interface returns $K_{\text{fail}} = H(z \parallel ct \parallel \texttt{04})$, rather than a key derived directly from $V_r$. An unstructured perturbation does not reveal the changed bit. To turn the returned key into a useful observation, an attack therefore requires a ciphertext that passes the activity filter and whose *other* verifier bits are already known. This differs from the raw key-mismatch interface used in several CPA-KEM and NIKE analyses [2, 4, 5, 7]. The following probe construction meets these requirements within HCR-KEM.

In this reference architecture, a unit-monomial core $u = bX^t e_\ell$, with coefficient (gain) $b$, activates all 128 quadratic CRT factors. In the source-admitted attack range, its bounded decoder inputs fix 289 non-target verifier positions to zero. The adversary changes one hint, computes the public confirmation hash for the guessed complete word, and compares the predicted accepted shared key with the ordinary 32-byte returned key. The resulting equality predicate yields a signed relation; a full-rank public graph certificate then determines the entire 1,536-coefficient state. The same graph also guides the query schedule: once cycle-closing relations are recovered, known parent coefficients allow five-valued searches in place of independent nine-valued searches. This is not a MAC forgery: the confirmation hash is unkeyed and the attacker constructs a matching tag for a guessed word. The ordinary-key recovery steps are summarized in Fig. 1.

**Contributions.** First, we specify HCR-KEM as a reference architecture isolating the difference between verifier-word confirmation and full ciphertext validation, and give a source-admitted ordinary-key comparison attack. Second, we prove deterministic isolation and signed-graph reconstruction of all 1,536 coefficients, with explicit ideal-model catalog and correctness bounds. Third, graph-aware peeling reduces online queries by using previously recovered coordinates to change the executable conditional threshold cuts. Fourth, we measure the original, depth-four scalar and graph-aware schedules on matched keys, and identify observable-interface boundaries for comparison with other KEMs. The detailed scalar amplitude frontiers and source-specific probability inventories appear in the technical sections and appendix.

**Table 1:** Contribution scope and validation boundary.

| Category | Verified or scoped contribution |
|---|---|
| Ideal-model | Public graph certificates, a restricted-history catalog-rank estimate, an honest decryption failure bound, and an attack-success advantage bound. |
| Compiled C | A 50-key matched evaluation across three query schedules: 150 successful recoveries, with mean graph-aware cost 3,565.18 calls. |
| Diagnostic and scope | Executable threshold-query analysis, a conditional independence criterion, and source-specific limits on cross-scheme conclusions. |

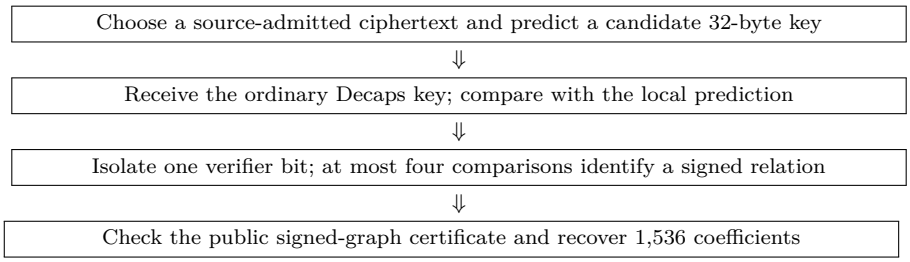


**Figure 1:** From a source-admitted probe to complete secret recovery. Only the ordinary 32-byte decapsulation key is returned; verifier bits and acceptance status are not exposed.

# 2 Preliminaries and Oracle Model

## 2.1 Confirmation and key derivation

Fix a reused secret decryption state $S$ and public key $pk$. A ciphertext has an unambiguous encoding

$$c = (u, \psi, d), \qquad \psi \in \mathcal{P}^L,$$

where $u$ is a core, $\psi$ a hint vector, and $d$ a confirmation tag. Let

$$V_S(u, \psi) \in \{0, 1\}^L$$

be a deterministic secret verifier. Public validation is $A(pk, u, \psi)$. Accepted decapsulation returns

$$P(pk, c, V_S(u, \psi))$$

exactly when validation succeeds and

$$d = T(pk, u, \psi, V_S(u, \psi)).$$

Otherwise it returns a failure key $\mathrm{KDF}_{\mathrm{fail}}(z, c)$. The attacker can compute $T$ and $P$ on a guessed verifier. No additional hidden check may reject a correctly guessed probe. The accepted and failure key functions use different random-oracle domains and output $\kappa$ bits; for the HCR-KEM instance, $\kappa = 256$ (32 bytes). The birthday estimate below uses the classical random-oracle model. Section 4.4.1 separately states deterministic implications for fixed hash functions.

For a guessed verifier $v$, define

$$\widehat{d} = T(pk, u, \psi, v), \quad \widehat{c} = (u, \psi, \widehat{d}), \quad \widehat{K} = P(pk, \widehat{c}, v).$$

The equality observation is

$$E(u, \psi, v) = \mathbf{1}[\mathrm{Decaps}(\widehat{c}) = \widehat{K}].$$

If the guess is correct and all checks pass, $E = 1$. For an incorrect verifier, equality requires a key-domain output collision even if the confirmation tags happen to collide. A confirmation-tag collision alone does not yield a correct predicted key.

**Lemma 1** (Whole-word comparison isolates one coordinate)**.** *Suppose a canonical, admitted core $u$ has a public baseline $\psi^0$ with known verifier vector $v^0$. Assume admission and all non-target verifier coordinates remain unchanged for every tested hint $h$, and that the guessed full-word confirmation tag passes whenever the target guess is correct. Changing only hint $r$ then leaves no uncertainty beyond its bit. One ordinary 32-byte key comparison determines that bit, except on an accidental key-equality event.*

*Proof.* Set $\psi = \psi^0[r \leftarrow h]$ and guess the known baseline vector with coordinate $r$ set to zero. If the actual coordinate is zero, the entire guessed verifier is correct and the candidate key is returned. If the coordinate is one, a different key-domain input is used or decapsulation takes the failure branch. Outside a key-equality collision, the comparison is unequal. Thus equality means zero and inequality means one. The argument fails if a different verifier coordinate changes or a hidden acceptance check rejects a correct guess. Section 4.1 establishes the non-target zero-arc condition, and Lemma 3 establishes complete selector initialization. □

Collisions of distinct confirmation-tag or key-domain inputs are charged to the full SHA3-256 collision event in Section 4.4. This matters because an accidental collision between different hash inputs could otherwise produce a false positive in the locally predicted-key equality test.

Let $\mathcal{D}$ be the set of distinct, fully encoded, domain-separated SHA3-256 inputs used in the confirmation-tag and returned-key derivations by the adversary and decapsulator during this experiment, and set $M = |\mathcal{D}|$. Each distinct input appears once, even if its hash is recomputed. The public attack transcript and all owner-side evaluations contribute to $M$; thus $M$ need not equal the number of Decaps calls. The conservative random-oracle birthday collision bound over these inputs is

$$\varepsilon_{\mathrm{eq}} \;\leq\; \frac{M(M-1)}{2^{\kappa+1}}. \tag{1}$$

An abstract bound on incorrect equality observations can replace (1). This birthday estimate must be evaluated with the distinct-input count for the chosen attack history, not inferred directly from the number of oracle calls.

**HCR-KEM instance in brief** The reference state is $(s, x)$ with $2kn = 1{,}536$ coefficients in $\{-2, -1, 0, 1, 2\}$, where $(n, k, q) = (256, 3, 3329)$. The selector forms a 290-bit verifier word, stored in 37 bytes. A source-admitted unit-monomial probe fixes 289 non-target bits, leaving one target hint for the comparison in Lemma 1. Table 2 gives the complete construction parameters; Section 4.1 proves the isolation property.

# 3 The HCR-KEM Construction

## 3.1 Target and representation

The accompanying project specification defines the source-available HCR-KEM research construction, with 29 forest shifts, ten slots per shift, and confirmation-tag validation. The following deterministic routines specify its measured C interface. Let $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^{256} + 1)$, with $q = 3329$ and module rank $k = 3$. Ring expressions below specify the intended modular operations. The implementation correspondence and compiler flags are recorded with the reproducibility materials.

Write $\mathrm{enc}_{12}$ for the source's coefficient packing, applied only after mapping coefficients to $\{0, \ldots, q-1\}$. Its decoder reports a noncanonical input if any 12-bit coefficient exceeds $q-1$; it initializes that coefficient to zero and continues. The encoder itself does not reduce negative integers. Sampled $\mathrm{CBD}_2$ coefficients have signed representatives in $\{-2, -1, 0, 1, 2\}$ and are stored as nonnegative residues in $\{3327, 3328, 0, 1, 2\}$ in `int16_t` coefficient arrays.

The public key is $pk = \rho \parallel \mathrm{enc}_{12}(t) \parallel \mathrm{enc}_{12}(\boldsymbol{p})$ and the secret key is $sk = \mathrm{enc}_{12}(s) \parallel \mathrm{enc}_{12}(x) \parallel z$. The ciphertext is $ct = (U, \Psi, d)$, where $U = \mathrm{enc}_{12}(u)$ and $\Psi$ packs the hint vector $\psi$. Capital letters distinguish encoded byte strings from algebraic objects. Throughout, $H$ denotes SHA3-256 on the displayed byte string; suffixes such as `02` denote one byte. Concatenation order is part of the specification.

**Table 2:** Parameters and serialized lengths.

| Quantity | Value | Meaning |
|---|---|---|
| $(n, k, q)$ | $(256, 3, 3329)$ | Ring and module |
| $\eta$ | 2 | Secret/noise CBD parameter |
| $J, C$ | $29, 10$ | Shifts and slots per shift |
| $L = JC$ | 290 | Reconciliation bits |
| $R_{\mathrm{rec}}$ | 779 | Conservative source radius |
| $\lvert pk\rvert, \lvert sk\rvert$ | 2336, 2336 bytes | Public and secret keys |
| $\lvert U\rvert, \lvert\Psi\rvert$ | 1152, 145 bytes | Core and hints |
| $\lvert d\rvert, \lvert ct\rvert$ | 32, 1329 bytes | Tag and ciphertext |
| $\lvert V\rvert, \lvert K\rvert$ | 37, 32 bytes | Packed word and key |

## 3.2 Sampling and public coordinate selection

The matrix expander uses SHAKE128 on $\rho \parallel i \parallel j$ for each of the nine entries of $A$. It processes the source's capped 1024-byte stream by 12-bit rejection sampling and fills any coefficient shortfall with zeros. The finite 1,024-byte stream cap is included explicitly in the matrix-expansion tail analysis of Section 4.5.2; the probability model does not assume an infinite rejection sampler. The vector and polynomial samplers use SHAKE256 on a 32-byte seed followed by a one-byte nonce, requesting 384 and 128 bytes, respectively. They map each nibble's four bits to $(b_0 + b_2) - (b_1 + b_3)$ modulo $q$.

For a core $U$, first compute $h_{\mathrm{ch}} = H(pk \parallel U)$, then request 256 bytes from $\mathrm{SHAKE256}(h_{\mathrm{ch}} \parallel \mathtt{01})$. The challenge sampler reads byte pairs, using the first byte for a shift and the low bit of the second for its sign. It retains the first occurrence of each shift. If fewer than 29 distinct shifts were found, it appends absent shifts in ascending order with sign $+1$. The result is $(\tau_j, \epsilon_j)_{j=0}^{28}$, with distinct $\tau_j$ and $\epsilon_j \in \{-1, +1\}$.

The forest selector uses union-find to retain only coordinate pairs that join distinct components, producing 290 acyclic selected positions on the 512-vertex bipartite topology. Its candidate scan and acceptance rule determine the public observation positions before any secret-dependent verifier bit is evaluated. It starts at shift $j = 0$ from 512 singleton vertices in one union-find structure; retained edges are never removed. At shift $j$, candidate $m$ proposes the edge $(m, 256 + (m - \tau_j \bmod 256))$; it is retained only when it joins different components. Each retry stream is SHAKE256 on $U \parallel j \parallel \mathit{retry} \parallel \mathtt{AA} \parallel \mathtt{F1}$, with $j$ and *retry* encoded as one byte each. For each $j \in \{0, \ldots, 28\}$, the raw selector allows $\mathit{retry} = 0, \ldots, 7$. Within each retry it first scans distinct byte values among the first 512 SHAKE256 output bytes, in order of first occurrence, and then scans the absent byte values $0, \ldots, 255$ in ascending order. It stops immediately upon ten accepted edges at that shift. Thus a complete retry-0 sweep covers all 256 candidates; under the distinct-shift forest invariant proved in Section 4.4.4 (Lemma 3), ten acceptances occur before that sweep is exhausted, and retries $1, \ldots, 7$ are not reached. Appendix A identifies the reference selector

implementation and its reproducibility requirements. Denote the selected coordinates by $m_{j,r}$. This selector uses $U$ and the shifts; it has no additional forest seed. Changing $\psi$ leaves every challenge and coordinate unchanged.

For sender or receiver polynomial inputs $a_\star, b_\star$, define the slot scalar

$$W_{j,r}(a_\star, b_\star) = [a_\star + \epsilon_j X^{\tau_j} b_\star]_{m_{j,r}} \pmod q.$$

The negacyclic wrap supplies its own sign in addition to $\epsilon_j$. The decoder selects the nearer of two circular centers; in particular, its zero arc is wide enough to keep the bounded non-target positions at zero. HelpRec on canonical $w$ sets $v = \lfloor B/16 \rfloor$ and $\psi = B \bmod 16$, where $B = \lfloor 32w/q \rfloor$. Rec chooses the nearer of the two centers

$$c_0(h) = \left\lfloor \frac{(2h+1)q + 32}{64} \right\rfloor = (2h+1)a,$$
$$c_1(h) = \left\lfloor \frac{(2h+33)q + 32}{64} \right\rfloor = (2h+33)a + 1$$

for $q = 64a + 1$ and $a = 52$, with $0 \le h < 16$, in circular distance, returning zero on a tie. Its zero arc is $[-780 + 104h, 884 + 104h] \pmod q$. Hints are packed two per byte, low nibble first. The 290 verifier bits are packed LSB first into $V$, with six zero padding bits. All 290 slots must be initialized before packing.

**Algorithm 1: Deterministic key generation**

1. Parse the supplied 64 bytes as $(d_{\text{seed}}, z)$, 32 bytes each.
2. Compute $(\rho, \sigma) = \text{SHA3-512}(d_{\text{seed}})$; expand $A$ from $\rho$.
3. Sample $(s, e, x, f)$ from $\sigma$ with vector-sampler nonces $(0, 1, 2, 3)$, respectively.
4. Set $t = As + e$ and $\boldsymbol{p} = Ax + f$ in $\mathcal{R}_q^3$.
5. Return $pk = \rho \parallel \text{enc}_{12}(t) \parallel \text{enc}_{12}(\boldsymbol{p})$ and $sk = \text{enc}_{12}(s) \parallel \text{enc}_{12}(x) \parallel z$.

**Algorithm 2: Deterministic encapsulation**

1. Take $pk$ and supplied 32-byte coins $\omega$; parse $pk$ as $(\rho, t, \boldsymbol{p})$ and expand $A$. There is no message-to-coins hash here.
2. Sample $r, e_1$ with vector-sampler nonces $0, 1$, seeded by $\omega$.
3. Set $u = A^T r + e_1$ and $U = \text{enc}_{12}(u)$.
4. Derive $(\tau_j, \epsilon_j, m_{j,r})$ as in Section 3.2.
5. Sample $e_2, e_3$ using polynomial-sampler nonces $2, 3$ and seed $\omega$. Compute $a_s = t^T r + e_2$ and $b_s = \boldsymbol{p}^T r + e_3$. Apply HelpRec to each $W_{j,r}(a_s, b_s)$. Pack the hints as $\Psi$ and the bits as $V_s$, including six zero padding bits.
6. Set $d = H(pk \parallel U \parallel \Psi \parallel V_s \parallel 02)$ and $ct = U \parallel \Psi \parallel d$.
7. Return $ct$ and $K_s = H(V_s \parallel pk \parallel ct \parallel 03)$.

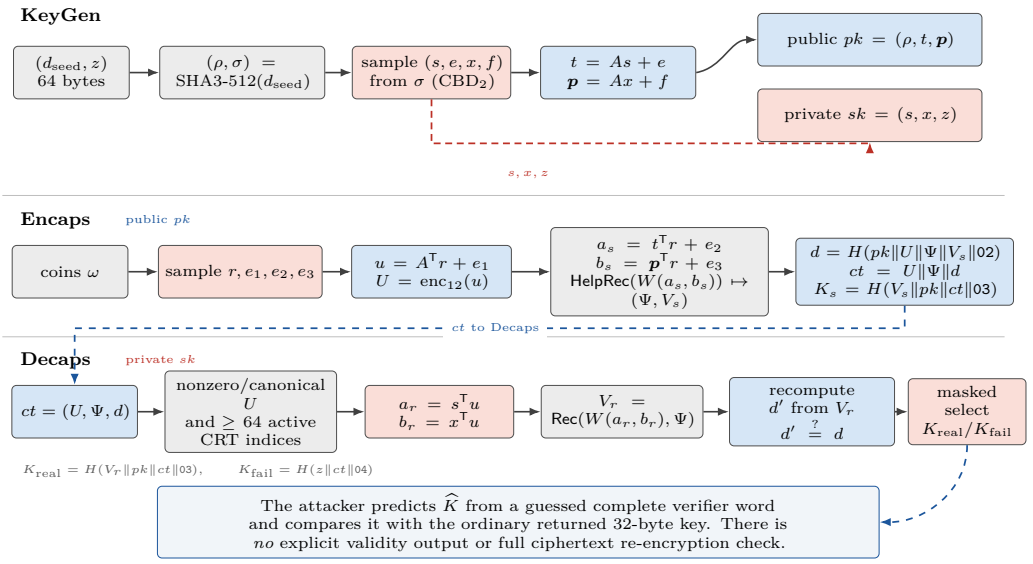


**Figure 2:** HCR-KEM key generation, encapsulation, and ordinary-key decapsulation. The public key contains $(\rho, t, \boldsymbol{p})$ and the secret key contains $(s, x, z)$. The public challenge selector is derived from $(pk, U)$; confirmation and masked key selection return one 32-byte key without an explicit validity bit or full re-encryption check.

**Algorithm 3: Standard decapsulation**

1. Parse $ct = (U, \Psi, d)$ and $sk$ as above. Decode $u, s, x$ and the 290 four-bit hints. Initialize $fail = 0$.
2. Set $fail = -1$ if $U$ is all zero, its 12-bit decoding is noncanonical, or fewer than 64 quadratic CRT indices have a nonzero factor in at least one of the three NTT-vector coordinates (as specified by the source admission rule). Continue.
3. Compute $a_r = s^T u$ and $b_r = x^T u$ using the dual inner-product routine with the precomputed transform of $u$.
4. Derive $(\tau_j, \epsilon_j, m_{j,r})$ from $pk, U$ as above.
5. Set $v^{\text{rec}}_{10j+r} = \mathsf{Rec}(W_{j,r}(a_r, b_r), \psi_{10j+r})$. Pack $V_r$ with six zero padding bits.
6. Compute $d' = H(pk \parallel U \parallel \Psi \parallel V_r \parallel \mathsf{02})$; set $fail = -1$ if $d' \neq d$.
7. Always compute both keys:

$$K_{\text{real}} = H(V_r \parallel pk \parallel ct \parallel \mathsf{03}),$$
$$K_{\text{fail}} = H(z \parallel ct \parallel \mathsf{04}).$$

8. Return $K_{\text{real}}$ if $fail = 0$, otherwise $K_{\text{fail}}$, using the source's masked selection. No acceptance bit is returned.

Algorithms 1–2 describe the deterministic interfaces. For the ideal-model claims, Algorithms 1–2 take independent uniform external coins; the reference convenience wrapper is not used as an entropy assumption. Algorithm 3 abstracts the standard key-return routine, not the auxiliary verifier-bit export routine.

Figure 2 summarizes the three interfaces and the ordinary key-return path.

# 4 Graph-Aware Peeling Cryptanalysis

The construction above exposes only a returned key, so the first step is to show how an admitted ciphertext makes that key informative. We then use the resulting relations to reconstruct the secret and reduce the number of comparisons.

## 4.1 One source-admitted probe yields one signed relation

Fix $q = 3329$, $n = 256$, $k = 3$, and the reference source identified in Appendix A. The state consists of the $D = 2kn = 1,536$ coefficients of $s, x$, each in $[-2, 2]$. Reference decapsulation rejects zero or noncanonical cores, fewer than 64 active quadratic CRT factors, or an invalid confirmation tag. It has no monomial-amplitude cap or reencryption check. The arithmetic interpretation is $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^{256} + 1)$, implemented by the pinned C routines.

For each $0 \le \ell < 3$, $0 \le t < 256$ and $104 \le b \le 195$, put $u = bX^t e_\ell$. There are $N_{\mathrm{cat}} = 3 \cdot 92 \cdot 256 = 70,656$ distinct, canonically encoded cores. We reserve $Q_{\mathrm{dec}} \le 6,144$ for online decapsulation comparisons; building the public catalog requires zero such queries. The nonzero entry $bX^t$ of each monomial vector is a unit of $\mathcal{R}_q$, so every quadratic CRT index has an active vector coordinate. For every selected expression, $|w| \le 4b \le 780$, inside the hint-zero arc $[-780, 884]$. All 290 baseline bits therefore equal zero under the source-model premises. Changing one designated hint leaves 289 non-target positions fixed, together with six zero padding bits. Full selector initialization is established in Section A.1.

### 4.1.1 The first actual occurrence and its signs

The reference challenge seed is $h = \text{SHA3-256}(pk \| U)$, without an additional domain byte. The first two bytes of $\text{SHAKE256}(h \| \texttt{01})$ give $\tau_0$ and $\epsilon_0 = (-1)^{\mathrm{byte}_1 \bmod 2}$. The first forest coordinate $m$ is the first byte of $\text{SHAKE256}(U \| \texttt{00 00 AA F1})$. The first shift candidate and first forest edge are always accepted. Write $t + \tau_0 = 256d + r$, $0 \le r < 256$, and set

$$\begin{aligned} i &= m - t \pmod n, \quad & j &= m - r \pmod n, \\ \alpha &= (-1)^{[m<t]}, & \beta &= \epsilon_0 (-1)^{d + [m<r]}. \end{aligned}$$

Then $o_{\mathrm{ori}} = \alpha$, $\epsilon = \alpha\beta$, and the measured integer is

$$w = b\, o_{\mathrm{ori}} (s_{\ell,i} + \epsilon x_{\ell,j}). \tag{2}$$

Indeed, the coefficient of $X^m$ in $X^t s_\ell$ has wrap sign $\alpha$. The shifted $x$ term first carries $(-1)^d$ from $X^{t+\tau_0}$ and then $(-1)^{[m<r]}$, giving $\beta$. This derives both endpoint signs from the actual source transcript. Negating a core does not establish a matching occurrence because it can change both selectors.

**Threshold geometry** For an oriented relation $v \in \{-4, \dots, 4\}$ and an admitted gain $b$, the source decoder realizes the ordered cut $v \ge c$ with

$$h(b, c) = \left\lfloor \frac{b(c-1) + 780}{104} \right\rfloor + 1, \qquad c = -3, \dots, 4. \tag{3}$$

The following branch illustrates two adjacent hint thresholds at the same public occurrence.

**One illustrative decision-tree branch** For a publicly derived row with orientation $o_{\mathrm{ori}} = +1$ and edge sign $\epsilon = +1$, suppose the two participating secret coefficients are both one. With an allowed gain $b = 104$, the hidden scalar is $v = 2$ and $w = bv = 208$. At $q = 3329$, the zero-bit threshold for hint $h = 9$ is 156, so $\mathrm{Rec}(208, 9) = 0$ and a predicted

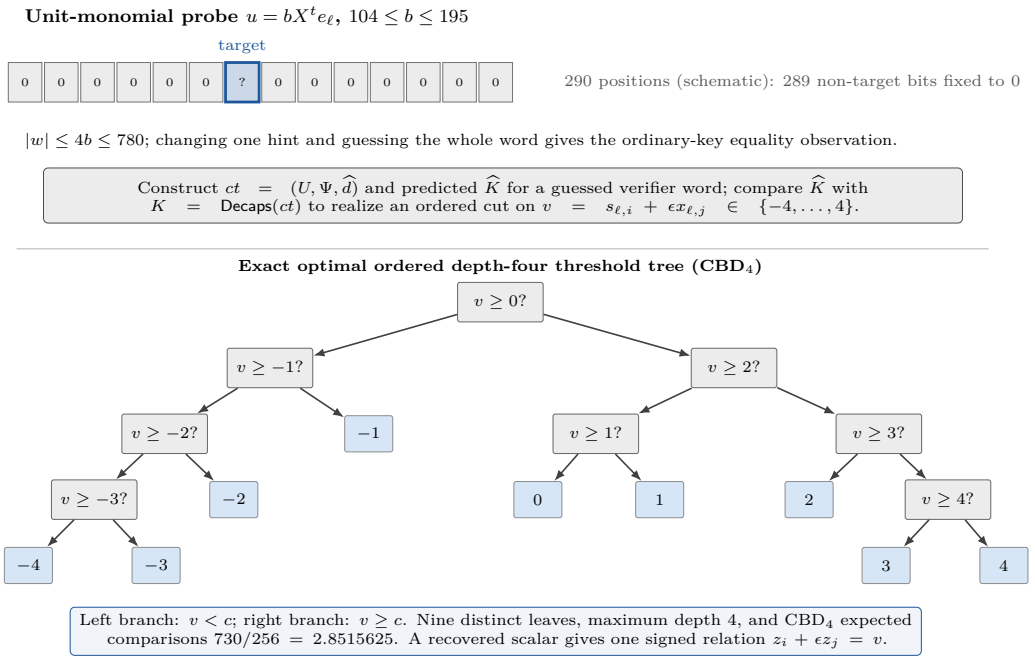


**Figure 3:** One source-admitted unit-monomial probe fixes 289 non-target verifier bits. The target hint gives an ordered cut through comparison with the ordinary returned key. The complete nine-leaf depth-four threshold tree identifies $v \in \{-4, \dots, 4\}$, with the CBD$_4$-weighted mean cost $730/256$.

all-zero verifier word yields a matching key under the stated assumptions. With $h = 10$ the threshold is 260 and $\mathrm{Rec}(208, 10) = 1$, so the same zero-word prediction mismatches. The remaining adaptive tests distinguish all nine possibilities $v \in \{-4, \dots, 4\}$ in at most four comparisons. Equality corresponds to target bit zero, not one.

For every adjacent pair of oriented values $v - 1, v \in \{-4, -3, \dots, 4\}$, the interval $((v-1)b, vb]$ has length at least 104 and lies in $[-780, 780]$. It contains a threshold from the odd-multiple grid. Lemma 5 (Appendix A.3.2) therefore supplies each required cut at this one occurrence. Four binary comparisons determine the oriented value; multiplication by $o_{\mathrm{ori}}$ gives the normalized row value. This larger-amplitude catalog avoids matching and does not replace the sharp fixed-gain optimum 92.

Fig. 3 illustrates the 290-position verifier strip, the returned-key comparison, and the exact ordered recovery tree.

The complete forest-initialization proof appears in Section A.1.

## 4.2 From a signed relation to state reconstruction

The public selector forest and the signed reconstruction graph are distinct; Fig. 4 illustrates their separate roles.

One recovered equation does not determine all coefficients. As a toy example, $s_0 + x_0 = 3$ leaves one degree of freedom, while the unbalanced closing relation $s_0 - x_0 = 1$ fixes $s_0 = 2$ and $x_0 = 1$. The full algorithm follows the same spanning-tree plus unbalanced closing-edge principle.

For one module component, let the graph have $2n$ vertices corresponding to

$$s_0, \dots, s_{n-1}, x_0, \dots, x_{n-1}.$$

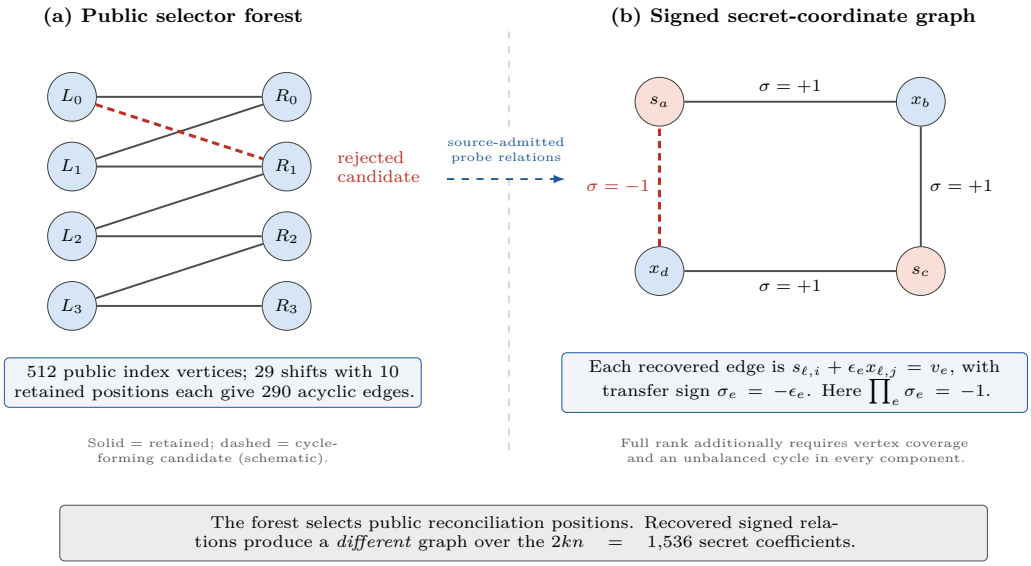


**Figure 4:** Two different graphs in HCR-KEM (schematic). The public selector forest retains 290 reconciliation positions on 512 index vertices. Source-admitted probe relations form a separate signed graph on $2kn = 1{,}536$ secret coefficients. Full-rank reconstruction requires complete vertex coverage and an unbalanced cycle in every component.

A matched row gives an edge between $s_i$ and $x_j$, with exact measured equation

$$z_{s_i} + \varepsilon_e z_{x_j} = v_e. \tag{4}$$

Give that edge transfer sign $\sigma_e = -\varepsilon_e$. A cycle is *unbalanced* when the product of its transfer signs is $-1$; otherwise it is balanced. These terms follow standard signed-graph theory, whose incidence-rank characterization is not new [11]. Fig. 5 illustrates the unbalanced-cycle initialization and subsequent signed-graph peeling.

**Lemma 2** (Exact graph rank and reconstruction)**.** *Over a field of characteristic different from two, the matrix of equations* (4) *has full column rank exactly when every connected component is unbalanced (so none is isolated). If this holds on $D$ covered vertices, a spanning tree and one unbalanced closing edge per component supply $D$ independent rows. Their consistent integer values reconstruct every vertex by rational arithmetic with only division by two at the root step.*

*Proof.* For a homogeneous system, an edge enforces $z_b = \sigma_e z_a$. Choose a root in a connected component. Every spanning-tree path expresses a vertex as $A_v z_r$, with $A_v \in \{+1, -1\}$. If all cycle products are positive, every value of $z_r$ satisfies all homogeneous equations, so the component has one-dimensional kernel. An unbalanced cycle instead enforces $z_r = -z_r$, hence $z_r = 0$ in characteristic different from two and then every vertex is zero. A graph's full matrix is block diagonal by connected components, proving the rank criterion.

For the inhomogeneous system, tree traversal computes

$$z_v = A_v z_r + B_v, \qquad A_v \in \{+1, -1\}, \quad B_v \in \mathbb{Z}.$$

For a traversal from vertex $u$ to vertex $v$, retain the original coefficients of the measured row:

$$c_{e,u} z_u + c_{e,v} z_v = v_e, \qquad c_{e,u}, c_{e,v} \in \{+1, -1\}.$$

For $s_i \to x_j$ these coefficients are $(1, \varepsilon_e)$; for $x_j \to s_i$ they are $(\varepsilon_e, 1)$. The direction-independent propagation formulas are

$$A_v = -c_{e,u}c_{e,v}A_u, \qquad B_v = c_{e,v}(v_e - c_{e,u}B_u).$$

Thus forward traversal gives $B_{x_j} = \varepsilon_e(v_e - B_{s_i})$, while reverse traversal gives $B_{s_i} = v_e - \varepsilon_e B_{x_j}$. Equivalently, swapping the endpoints in a row normalized to starting coefficient one requires multiplying its right-hand side by $\varepsilon_e$. For an unbalanced closing edge, substitution gives

$$(c_{e,u}A_u + c_{e,v}A_v)z_r = v_e - c_{e,u}B_u - c_{e,v}B_v.$$

Its left coefficient is $+2$ or $-2$, by unbalance. This fixes the root and therefore every vertex uniquely. Consistency with an actual integer secret makes the division exact. A tree has one fewer edge than vertices; one unbalanced closing edge gives exactly as many independent rows as vertices. Summing over components gives $D$ rows. □

This lemma uses *exact measured integers*. Full rank of rounded, interval-valued, or noisy rows alone would not justify the conclusion. Box constraints may sometimes resolve a balanced component, but no such additional recovery claim is needed here. The same certificate that establishes unique recovery also determines which relations need to be queried first.

## 4.3 Graph-Aware Peeling and Optimal Query Scheduling

Once the signed-graph certificate is known, recovering every relation independently is unnecessary. We first recover the relations on unbalanced cycles; a recovered parent coefficient then turns each remaining relation into a five-valued child observation. This changes the admissible conditional cuts rather than the independent coefficients' marginal entropy.

**Why threshold geometry reduces oracle calls** A normalized signed relation is the sum or difference of two $\mathrm{CBD}_2$ secret coefficients. Under independent honest secret sampling, the nine scalar values in $\{-4, \dots, 4\}$ have the $\mathrm{CBD}_4$ weights $(1, 8, 28, 56, 70, 56, 28, 8, 1)/256$. The actual source decoder permits the threshold cut $v \geq c$ for $c = -3, \dots, 4$ by choosing any admitted gain $b \in [104, 195]$ and the hint $h(b, c)$ defined in (3). A compiled-source test of the decoder passed all $92 \times 8 \times 9 = 6{,}624$ gain/cut/scalar cases. A depth-four ordered decision tree optimal among depth-four ordered trees for independent nine-valued scalar identification has expected weighted path length $730/256 = 2.8515625$ tests per scalar, giving 4,380 expected comparisons across 1,536 recovered relations under that marginal model.

**Admissible binary recovery trees** Let $\mathcal{T}$ denote the transcript under a reused-secret oracle and $\pi_{\mathcal{T}}$ the posterior on the remaining secret states. Only actual source-admissible queries belong to $\mathcal{Q}(\mathcal{T})$, including the ciphertext-validity restrictions and the *returned* oracle output. For unit-cost deterministic binary queries, the minimum expected remaining number of observations is

$$\mathsf{Cost}(\mathcal{T}) = \min_{q \in \mathcal{Q}(\mathcal{T})} \Big\{ 1 + \sum_{y \in \{0,1\}} \Pr[y \mid \mathcal{T}, q]\, \mathsf{Cost}(\mathcal{T}, q, y) \Big\}. \tag{5}$$

We set $\mathsf{Cost}(\mathcal{T}) = 0$ when $|\operatorname{supp} \pi_{\mathcal{T}}| = 1$. The minimum considers informative queries; if no admissible continuation separates the remaining states, set $\mathsf{Cost}(\mathcal{T}) = \infty$. If costs differ,

replace the leading one by $c(q)$. When query availability depends on the posterior alone, write $\mathsf{Cost}(\pi)$. For noisy outputs use the corresponding likelihoods.

Entropy bounds the expected number of binary identifications from below. Unrestricted Huffman coding minimizes length over all binary partitions, but its nodes need not correspond to executable ciphertext queries. Ordered-threshold binary recovery trees impose further restrictions. Thus the optimum for a fixed scalar observation family cannot be assumed globally optimal against joint multi-positional strategies [4–9].

**Table 3:** Exact expected binary comparisons. Huffman permits arbitrary partitions; ordered columns permit only threshold cuts.

| Prior | Huffman | Ordered | Ordered depth cap |
|---|---|---|---|
| $\mathrm{CBD}_2$ | 34/16 | 37/16 | 37/16 ($d \le 3$) |
| $\mathrm{CBD}_4$ | 662/256 | 697/256 | 730/256 ($d \le 4$) |

The Shannon entropies of $\mathrm{CBD}_2$ and $\mathrm{CBD}_4$ are 2.030639 and 2.544198 bits, respectively. The recurrence below gives $D(0, 8, 4) = 730$. Removing the depth-four restriction gives weighted optimum 697, attained at maximum depth five (a nominal $697/256 = 2.72265625$ queries per relation, or 4,182 for 1,536 relations). This gives a second, unrestricted ordered reference point for the cost model. The measured graph-peeling comparison uses the implemented depth-four tree. For $\mathrm{CBD}_2$ the depth-three ordered optimum is 37/16, compared with the unrestricted Huffman lower bound 34/16. This distinction matters because the graph method changes the admissible conditional observation problem, rather than only reordering the same nine-value scalar queries. The observed HCR scalar comparator is therefore optimal *within the executable depth-four ordered threshold class*, not among all conceivable adaptive oracle strategies.

**Optimality certificate for the threshold tree** Let $w_i = \binom{8}{i}$ be the unnormalized $\mathrm{CBD}_4$ weight of $i - 4$, and $W_{\ell,r} = \sum_{i=\ell}^{r} w_i$. For an ordered threshold decision tree restricted to maximum depth $d$, let $D(\ell, r, d)$ be the minimum weighted sum of remaining comparisons on outcomes $\ell, \ldots, r$. Each comparison adds $W_{\ell,r}$ to the objective, and a threshold splits the interval at one $k$. Consequently

$$D(\ell, r, d) = W_{\ell,r} + \min_{\ell \le k < r} \big\{ D(\ell, k, d-1) + D(k+1, r, d-1) \big\}.$$

Set $D(i, i, d) = 0$ and $D(\ell, r, 0) = \infty$ when $\ell < r$. Induction over interval size and depth makes this recurrence an optimality certificate for all ordered depth-four trees. Its minimizing tree begins with cut $c = 0$; the left subtree uses $-1, -2, -3$, while the right uses 2 followed by $1, 3, 4$. Evaluating the integer recurrence gives $D(0, 8, 4) = 730$. The model expectation is 730/256 comparisons per relation; the experiment measures calls on actual public selectors and sampled secret keys.

**Graph-conditioned extraction** Figure 5 illustrates why conditioning on a known parent permits a smaller ordered recovery tree.

For each certified three-component signed graph, let $C_{\mathrm{cyc}}$ be the number of distinct edges on its three fundamental unbalanced cycles, including their closing edges. We first recover those edge relations with the nine-value depth-four tree and solve all cycle vertices. Every other forest edge then has a known parent coefficient and an unknown child in $\{-2, \ldots, 2\}$. Subtracting the parent reduces its observation to five ordered outcomes, resolved by a depth-three tree. Applying the same recurrence with $\mathrm{CBD}_2$ weights $(1, 4, 6, 4, 1)$ gives

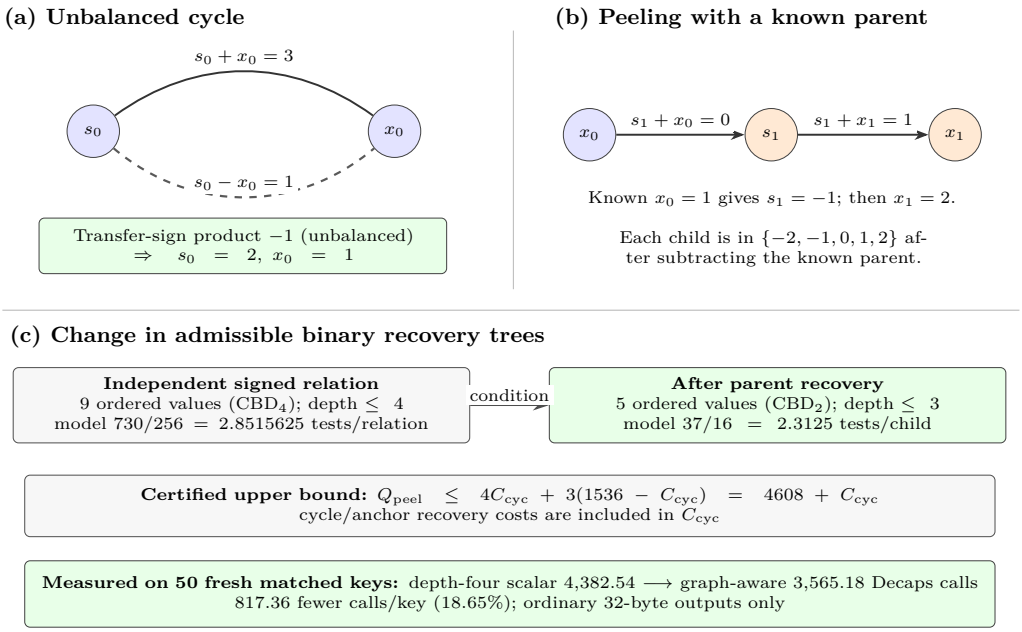


**Figure 5:** Signed-graph initialization and graph-aware peeling. The schematic unbalanced two-edge cycle determines an initial coefficient pair; each recovered parent then reduces a remaining child search from nine to five possible values. The bound includes cycle-edge costs, and the matched 50-key means are ordinary 32-byte Decaps query counts.

exact optimal weighted cost 37/16 for this restricted child query. Thus a valid source basis permits at most

$$Q_{\mathrm{peel}} \leq 4C_{\mathrm{cyc}} + 3(1536 - C_{\mathrm{cyc}}) = 4608 + C_{\mathrm{cyc}}.$$

Although box-constrained sparse solvers may exploit underdetermined systems, the reported method prioritizes exact signed-graph closure with an independently checkable reconstruction certificate. Such alternative post-processing was not evaluated.

**Observation support and conditional cuts** Let $\mathcal{V}$ index the secret-state coordinates. The support $e(q) \subseteq \mathcal{V}$ of a permissible query $q$ is the smallest set of coordinates on which its *actual observable* depends after the transcript and public validation are fixed. The hypergraph is $\mathcal{H} = (\mathcal{V}, \{e(q) : q \in \mathcal{Q}\})$. An operation combining four reconciliation residuals need not produce a four-vertex secret-coordinate hyperedge: each residual may itself combine many secret coefficients.

For the pinned HCR source, source-admitted probes provide exact relations $z_i + \epsilon_e z_j = v_e$. A signed-graph certificate first resolves unbalanced cycles. A known parent then converts the remaining signed-pair relation into a five-valued child observation. The smaller tree is available because of the *observation geometry*, rather than an automatic reduction in marginal entropy: for independent coefficients, $H(z_j \mid z_i) = H(z_j)$ before observing the adaptive transcript.

For general schemes, an improvement requires executable conditional cuts after public validation and a cost advantage over the best strategy for the *same* oracle. The cycle and anchor costs must be included. No conclusion about external lattice schemes follows from their ring or reconciliation formulas alone.

## 4.4 Reconstruction and Pre-Challenge Security

### 4.4.1 Public certificate and deterministic extraction

Write $\mathsf{CertValid}_{\mathrm{spec}}(pk)$ when the exact-arithmetic first-occurrence catalog of Section 4.1 admits a publicly verified full-rank signed-graph basis. A witness contains one spanning tree and one unbalanced closing edge per component; the verifier checks core membership, derived endpoints and signs, component coverage, acyclicity, and closing-edge identities. The certificate is public and checkable from the canonical catalog. The whole-word comparison of Lemma 1 supplies individual signed relations; the certified full-rank graph of Lemma 2 then determines the complete state.

**Theorem 1** (Certificate dichotomy and deterministic extraction)**.** *For each public key, exact catalog processing yields either a full-rank signed-graph certificate or a nonzero ternary vector annihilated by all catalog rows. Both are publicly checkable. Under exact ring arithmetic in Algorithms 1–3,* $\mathsf{CertValid}_{\mathrm{spec}}(pk)$ *implies recovery of* $(s, x) \in \{-2, -1, 0, 1, 2\}^{1536}$ *using at most 6,144 pre-challenge ordinary key-return comparisons, outside distinct-input SHA3-256 output collisions. In this experiment* $\delta_{\mathrm{recovery,spec}} = 0$*.*

*Proof.* A balanced graph component supplies a nonzero homogeneous $\{0, \pm 1\}$ kernel by propagation along its tree; otherwise Lemma 2 supplies the certified full-rank basis. $\mathrm{CBD}_2$ puts all secret coefficients in the displayed alphabet. Every $bX^t e_\ell$ catalog core is a nonzero canonical unit: all 128 quadratic CRT factors are active, exceeding the admission threshold of 64. Lemma 3 initializes all 290 selected coordinates on retry zero. For every selected scalar, $|w| \le 4b \le 780$ is inside the hint-zero arc $[-780, 884]$. One changed hint leaves the other 289 verifier bits and the six packed padding bits fixed. The correctly predicted full word and tag pass the specified confirmation check; an incorrect guess returns an unequal 32-byte key except on a counted collision. The depth-four tree identifies each signed integer relation in at most four comparisons. The rational signed-graph reconstruction of Lemma 2 uniquely fixes all 1,536 coordinates. There is thus no further probabilistic extraction step. □

### 4.4.2 Local prediction and a quantified attack-success experiment

After the allowed pre-challenge queries, the adversary encodes recovered $(\widehat{s}, \widehat{x})$ canonically, appends a public $z_0 = 0^{256}$ and runs Algorithm 3 locally on the honest challenge $ct^*$, producing $\widehat{K}$. No target decapsulation query occurs after the challenge. Let $R$ mean exact recovery, $A$ honest acceptance, and $M$ equality of the sender and receiver packed verifier words. On $R \cap A \cap M$ the genuine and predicted accepted keys both equal $H(V_r \parallel pk \parallel ct^* \parallel \mathtt{03})$; the fallback field $z$ is irrelevant. With independent random $K_{\mathrm{rand}} \in \{0,1\}^{256}$, the equality-test advantage is $\Delta = \Pr[\widehat{K} = K_s] - 2^{-256}$.

The following game is an idealization of the specific HCR-KEM research construction described in Section 3, not a general security theorem about confirmation-tagged KEMs. Fix a particular ideal-oracle experiment: uniform independent 64-byte KeyGen and 32-byte Encaps coins; independent SHA3-256, SHA3-512, SHAKE128 and SHAKE256 random interfaces with equal-input reuse; and the restricted history of Section 4.5.1, with catalog preprocessing before the honest challenge and no extra direct adversarial SHA3-256 calls. There are $Q_{\mathrm{cat}} = 70,656$ catalog entries, at most 6,144 oracle Decaps comparisons, and at most $N_{H,\mathrm{full}} = 107,531$ SHA3-256 finalizations, counted as in Section 4.4.4 with $q_{\mathcal{A}} = 0$. For arbitrary additional oracle queries these budgets must be increased; the numerical bound below concerns this explicit adversary, not all possible oracle histories.

**Theorem 2** (Ideal-model attack success and challenge-key prediction)**.** *Let* $B_{\mathrm{rank}} = (1 + 2e^{-1403/16})^{512} - 1 + 3^{512} e^{-759}$ *and* $\delta_{\mathrm{seed,cat}} \le Q_{\mathrm{cat}}(Q_{\mathrm{cat}} - 1)2^{-257} + Q_{\mathrm{cat}} 2^{-256}$*. In*

**Table 4:** Conservative SHA3-256 finalization inventory for the specified restricted-history attack. Repeated identical inputs may be counted more than once.

| Phase | Bound |
|---|---|
| Catalog challenge digests | 70,656 |
| Ordinary Decaps ($4 \times 6,144$) | 24,576 |
| Predicted tags and keys ($2 \times 6,144$) | 12,288 |
| Recovery bookkeeping | 4 |
| Honest encapsulation and local prediction | 7 |
| Complete experiment | 107,531 |

*the exact-arithmetic restricted-history ideal experiment, with the honest-correctness bound* $\delta_{\mathrm{cor}} < 2^{-119}$ *of Theorem 3,*

$$\begin{aligned}\Delta_{\mathrm{model}} \geq 1 - \big(&3B_{\mathrm{rank}} + \delta_{\mathrm{seed,cat}} + \delta_{\mathrm{cor}} \\ &+ \tbinom{107531}{2}2^{-256} + 2^{-256}\big) > 1 - 2^{-114}.\end{aligned} \tag{6}$$

*This is an existence-of-adversary statement for the stated idealized IND-CCA1 experiment, not a bound for an unrelated deployed implementation.*

*Proof.* The restricted-history catalog estimate, Lemma 4, gives rank failure probability at most $3B_{\mathrm{rank}} + \delta_{\mathrm{seed,cat}}$. Rank implies $\mathsf{CertValid}_{\mathrm{spec}}$; Theorem 1 then gives exact extraction except on distinct-input output collisions. Theorem 3 bounds honest challenge rejection and verifier-word disagreement. The locally predicted key matches the genuine accepted key outside the union of these events. The SHA3-256 collision term is at most $\binom{107531}{2}2^{-256}$ and is charged once, including candidate tags and fallback keys. An independent uniform key matches the local prediction with probability exactly $2^{-256}$. Applying the union bound and the Section 4.5.1 evaluation $3B_{\mathrm{rank}} \approx 2^{-114.92136}$ gives the inequality. □

For the specified adversary, catalog construction evaluates at most $N_{\mathrm{cat}} = 70,656$ challenge digests; a single chosen-hint comparison computes one candidate confirmation tag and one candidate accepted key, hence $L \leq 2Q$ client-side SHA3-256 calls. Ordinary Decaps computes four further SHA3-256 outputs per call, with $Q \leq 6,144$. One honest challenge encapsulation and one local post-challenge Decaps contribute three and four calls, respectively. Allowing four additional fixed recovery bookkeeping hashes gives the complete conservative inventory in Table 4, totaling $N_{H,\mathrm{full}} \leq 107{,}531$.

The fixed adversary does not make other direct SHA3-256 queries. Repeated evaluation of identical inputs is counted conservatively. The catalog's public graph calculation uses no target Decaps calls; the extraction tree's query cap is independent of its public preprocessing cost. The same fixed history supplies the catalog-rank estimate and the honest-coin coupling bound $B_{33} \leq 76{,}801$.

The mathematical model and the compiled-source experiments have different verification obligations, summarized in Section 5.5.

### 4.4.3 Validity-check countermeasure

HCR-KEM authenticates the verifier word without re-encrypting and comparing the full ciphertext. FIPS 203 Algorithm 18 instead performs complete re-encryption comparison [19]. Adapting such a transform requires a newly specified message, coins derivation, validity check and security proof, not merely a longer confirmation tag. SMAUG-T and NTRU+ use distinct validation mechanisms [21, 22].

**Proposition 1** (Validity check against this probe family)**.** *Suppose a specified repaired decapsulator sends invalid probes to the fallback branch and an adaptive family contains a*

*valid probe with probability at most* $\varepsilon_{\text{valid}}$*. The probability of a positive predicted-real-key comparison is at most* $\varepsilon_{\text{valid}} + \Pr[\mathsf{Bad}_{\text{cross}}]$*, where the latter event counts collisions across fallback and accepted-key outputs.*

*Proof.* Fallback and accepted-key inputs have distinct final domain bytes `04` and `03`. On any invalid probe, equality with a predicted real key requires a SHA3-256 output collision between different inputs; otherwise the probe must pass the validity check. Union-bounding these events proves the claim. □

This proposition addresses the displayed observation channel, not full CCA security of an unspecified repair.

#### 4.4.4 Exact Forest Finalizations and Complete-Transcript Inventory

**Lemma 3** (Forest finalizations under the documented selector)**.** *For 29 pairwise distinct shifts indexed* $j = 0, \ldots, 28$*, the documented two-phase retry-*0 *scan, starting with 512 singleton vertices, completes ten slots per shift. The specified algorithm therefore takes retry* 0 *only and uses* $N_{\text{forest}} = 29$ *SHAKE256 finalizations. Applying the specification lemma to a separately compiled executable requires source and build correspondence.*

The proof, including first-shift initialization, appears in Appendix A.1.

**Table 5:** Finalizations per reference routine, with distinct forest shifts. Squeezed blocks are not additional finalizations.

| Routine | SHA3 256 | SHA3 512 | SHAKE 128 | SHAKE 256 |
|---|---|---|---|---|
| Key generation | 0 | 1 | 9 | 4 |
| Encapsulation | 3 | 0 | 9 | 34 |
| Decapsulation | 4 | 0 | 0 | 30 |

The four decapsulation SHA3-256 finalizations are the challenge digest in step 4, the tag in step 6, and both keys in step 7. Failure does not skip either key finalization. The challenge sampler specified in Section 3.2, used by both Algorithms 2 and 3, derives 29 distinct shifts even on adversarial ciphertexts: the sampler keeps a seen set and fills any remaining positions with unseen indices. Caller-supplied-shift helper variants are excluded. For one key generation, $Q$ standard target decapsulations, one honest challenge encapsulation and at most one standard local prediction, the KEM routines alone contribute

$$N_{256}^{\text{KEM}} \le 4Q + 7, \qquad N_{512}^{\text{KEM}} = 1,$$
$$N_{\text{XOF128}}^{\text{KEM}} = 18, \qquad N_{\text{XOF256}}^{\text{KEM}} \le 30Q + 68.$$

At $Q = 6144$, the last bound is 184388 under this standard-routine scope; arbitrary-shift helpers retain only $N_{\text{forest}} \le 232$. Catalog generation, client-side candidate-key calculations and direct adversarial calls are additional work.

For SHA3-256, a recovery inventory of $N_{\text{cat}}$ catalog calls, $Q$ target calls, $L$ local candidate-key calls and $E_{\text{rec}}$ other calls yields

$$N_{H,\text{full}} \le N_{\text{cat}} + 4Q + L + E_{\text{rec}} + 3 + 4 + q_{\mathcal{A}}. \tag{7}$$

Here $q_{\mathcal{A}}$ counts only direct calls not already charged, and all calls made by the chosen predictor must be included. If a separately verified inventory establishes $N_{\text{cat}} \le 70{,}656$, $L \le 2Q$, $E_{\text{rec}} \le 4$ and $Q \le 6144$, this gives $N_{H,\text{full}} \le 107531 + q_{\mathcal{A}}$. The corresponding $107524 + q_{\mathcal{A}}$ excludes the challenge and local prediction and is a pre-challenge extraction

bound. Neither number can be assigned to a different predictor without recounting its calls.

If SHA3-256 is modeled as one random oracle and all relevant distinct-input collisions are covered, a deterministic full-game cap $N$ gives $\Pr[\mathsf{Bad}_{\mathrm{full}}] \le \binom{N}{2}/2^{256}$. This does not model the remaining primitives. A complete ideal-model theorem must explicitly specify SHA3-512, SHAKE128 and SHAKE256 interfaces, preserve equal-input reuse within each, and count all their catalog and predictor work. The inventory table alone is no proof about concrete Keccak security or about independence of values selected adaptively through the public key.

## 4.5 Ideal-Model Availability and Honest-Correctness Proofs

### 4.5.1 Public Catalog Availability Bound

The catalog estimate applies in the restricted-history ideal-oracle experiment specified in Section 4.4. Its precise fresh-stream and coin assumptions follow.

For this section, SHA3-256, SHA3-512, SHAKE128 and SHAKE256 are independent ideal interfaces of the indicated output sizes. Distinct inputs to the same XOF have independent streams; equal-length inputs are not assumed domain-separated merely because they serve different purposes. Key generation uses uniformly supplied coins and has no earlier oracle history. Condition on its transcript. Before catalog preprocessing, there are no previous queries to the catalog's challenge-hash or forest inputs.

In the restricted ideal-oracle history, $Q$ distinct challenge-hash inputs yield independent uniform 256-bit seeds unless an accounted input or seed collision occurs. These seeds can collide with one another or with the seed prefix of a previous relevant 33-byte SHAKE256 input ending in `01`. With $q_X$ such previous inputs,

$$\delta_{\mathrm{seed,cat}} \le \frac{Q(Q-1)}{2^{257}} + \frac{Qq_X}{2^{256}}.$$

In the key-generation-only history $q_X = 1$. Outside this event the first $(\tau_0, \epsilon_0, m)$ values are fresh independent outputs for distinct cores. For each module the $N = 92n$ normalized rows are therefore independent uniform-endpoint, unbiased-sign rows $e_i + \epsilon e_{n+j}$. The first forest input has $1,156$ bytes and does not alias these 33-byte challenge-XOF inputs. The conditioning concerns seed values and old input names; it does not expose the fresh XOF output bytes.

**First-occurrence row distribution** We make the distributional bridge explicit. For each fixed catalog core $u = bX^t e_\ell$, the first two fresh challenge-XOF bytes give $\tau_0 \in \{0, \dots, 255\}$ uniformly and an independent unbiased orientation sign $\epsilon_0$. The first forest-XOF byte gives a uniform $m \in \{0, \dots, 255\}$, independently of $(\tau_0, \epsilon_0)$: the forest's initial union-find has only singleton components, so its first proposed edge is accepted. Freshness follows from the preceding seed-collision exclusion. Challenge inputs have length 33 bytes and forest inputs length 1,156 bytes; the latter are pairwise distinct across catalog cores because their canonical encoded $U$ fields are distinct.

For fixed $t$, the two graph endpoints are

$$\begin{aligned} i &= m - t \pmod{256}, \\ j &= m - (t + \tau_0 \bmod 256) \pmod{256}. \end{aligned}$$

The transformation $(m, \tau_0) \mapsto (i, j)$ is bijective: its inverse is $m = i + t \bmod 256$ and $\tau_0 = i - j \bmod 256$. Thus $(i, j)$ is uniform on the full $256 \times 256$ bipartite endpoint set. The negacyclic wrap and source challenge orientation determine an edge sign $\epsilon = \alpha\beta$, where $\beta = \epsilon_0(-1)^{d+[m<r]}$ and $\alpha = (-1)^{[m<t]}$ as in Section 4.1. For fixed $(i, j, t)$, the

parity factors are fixed and the independent fair $\epsilon_0$ makes $\epsilon$ an unbiased sign. Hence every normalized row is distributed exactly as $e_i + \epsilon e_{256+j}$ with independent uniform endpoints and an independent fair sign.

The same argument holds jointly over distinct catalog cores: their challenge-XOF and forest-XOF inputs are distinct, and fresh ideal-XOF outputs are independent under the restricted history. Therefore each module's $92 \cdot 256$ first-occurrence rows satisfy the independent-row law used in Lemma 4. No independence between the three modules' *rank-failure events* is required for the final union bound. This establishes the selector-to-rank-model bridge only within the specified ideal interfaces and history.

**Lemma 4** (Support-sensitive catalog rank bound)**.** *For $n = 256$ and $N = 92n$, a module's rank-failure probability in the fresh-output experiment is at most*

$$B_{\mathrm{rank}} = \left(1 + 2e^{-1403/16}\right)^{512} - 1 + 3^{512} e^{-759}.$$

*Consequently $\delta_{\mathrm{cert}} \le 3B_{\mathrm{rank}} + \delta_{\mathrm{seed,cat}}$.*

*Proof.* The signed-graph rank criterion implies that failure gives a nonzero homogeneous-kernel vector in $\{0, \pm 1\}^{2n}$: choose one balanced component, propagate its signs, and put zero elsewhere. This includes an isolated vertex. Fix such a vector, with $l$ left and $r$ right nonzero entries and $m = l + r$. One row violates it with exact probability

$$p_{l,r} = \frac{l+r}{n} - \frac{3lr}{2n^2}.$$

Exactly one supported endpoint always violates; two supported endpoints violate with probability one half. Since $lr \le m^2/4$,

$$p_{l,r} \ge \frac{m}{n} - \frac{3m^2}{8n^2}.$$

For $1 \le m \le 32 = n/8$, this is at least $61m/(64n)$; $N$ independent rows thus annihilate the vector with probability at most $e^{-1403m/16}$. There are $\binom{512}{m}2^m$ choices at support $m$. Extending their positive sum gives at most $(1+2e^{-1403/16})^{512} - 1$. For $m \ge 33$, the general bound $p_{l,r} \ge m/(4n)$ holds because $m \le 2n$. Survival is at most $e^{-23m} \le e^{-759}$. There are fewer than $3^{512}$ ternary vectors, giving the second term. Union-bound the two support ranges, then the three modules, and finally the seed event. No independence between module-failure events is needed. □

Numerically, the stated expression gives $3B_{\mathrm{rank}} \approx 2^{-114.92} < 2^{-114}$; the separate $\delta_{\mathrm{seed,cat}}$ term must also be added. This is an ideal-model catalog-rank estimate under the restricted query history, not by itself a lower bound on the full-extraction event $\Pr[R]$ defined in Section 4.4.2. The refined estimate uses the actual support dependence of each catalog event. The signed-graph rank theorem is standard; the displayed finite-parameter probability estimate is derived here for the catalog.

### 4.5.2 Honest Decapsulation Correctness

The honest-correctness bound is needed for the locally predicted honest challenge key to match the legitimate encapsulated key once the reused state has been recovered. This is separate from the canonical admission and whole-word assumptions used for the attack probes. We bound honest decapsulation failure in the same explicitly specified ideal-oracle model using the actual decoder, noise law and finite rejection tests.

**Theorem 3** (Ideal-model decapsulation correctness bound)**.** *For the inspected arithmetic and uniformly supplied ideal-sampling coins, the probability of honest decapsulation failure, averaged over KeyGen and one Encaps, is less than* $2^{-119}$*. This includes structural rejection and the finite matrix-expansion cap. The estimate remains valid for the restricted-history ideal-oracle catalog experiment with the fresh-coin coupling below.*

The proof controls all possible labels before applying the selector, so no independence between the selected label and the noise is required. The probability is not silently conditioned on a chosen public key or on successful certification.

**Source binding and experiment** The pinned reference routines implement key generation, encapsulation, decapsulation, polynomial sampling, matrix expansion, the activity filter and reconciliation. Source identities and code-level correspondence are documented in the accompanying artifact.

Assume these arithmetic functions implement the specified ring $\mathcal{R}_q = \mathbb{Z}_{3329}[X]/(X^{256}+1)$. Take $n = 256, k = 3, a = 52$. Initially sample all coefficients of $s, e, x, f, r, e_1, e_2, e_3$ independently from CBD2, and take the matrix uniform in $\mathcal{R}_q^{3\times 3}$. The paragraphs below handle the finite matrix-expansion cap and the ideal-oracle seed coupling. The convenience `rand()` wrapper is not the source of theoretical entropy.

This source has no ciphertext compression error in $u$. The sender computes

$$\begin{aligned} t &= As + e, & \boldsymbol{p} &= Ax + f, \\ u &= A^T r + e_1, & a_s &= t^T r + e_2, \\ b_s &= \boldsymbol{p}^T r + e_3. \end{aligned}$$

The receiver computes $a_r = s^T u$, $b_r = x^T u$. Thus the exact integer noise polynomials, reduced modulo $q$ only after the identity, are

$$\begin{aligned} E_a &= a_s - a_r = e^T r - s^T e_1 + e_2, \\ E_b &= b_s - b_r = f^T r - x^T e_1 + e_3. \end{aligned}$$

The matrix terms cancel for every matrix. Consequently the reconciliation estimate below does not itself require uniform matrix sampling.

For a label $(\tau_0, \epsilon_0, m)$, $0 \le \tau_0, m < n$, $\epsilon_0 \in \{\pm 1\}$, let

$$\Delta_{\tau_0,\epsilon_0,m} = [E_a + \epsilon_0 X^{\tau_0} E_b]_m.$$

This is exactly the difference between the sender and receiver values at that label, including the negacyclic sign.

**Guaranteed decoding radius** Let $B = \lfloor 32w/q \rfloor$, $v_B = \lfloor B/16 \rfloor$, $h = B \bmod 16$. The sender's integer bin is

$$I_B = [\lceil Bq/32 \rceil,\ \lceil (B+1)q/32 \rceil - 1].$$

For $q = 64a + 1$, $I_0 = [0, 2a]$, while for $1 \le B \le 31$, $I_B = [2aB + 1, 2a(B+1)]$. The receiver's zero arc is

$$Z_h = [(2h-15)a, (2h+17)a] \pmod q,$$

with its endpoints included. The one arc is its integer complement, $[(2h+17)a+1, (2h+49)a] \pmod q$.

For $B = h < 16$, extending $I_B$ by $15a$ in both directions stays in $Z_h$. For $B = h+16$, the corresponding extension stays in the one arc. This verifies the exact endpoint convention, including $B = 0$ and the modular boundary. Hence error at most $15a = 780$ is sufficient.

We use the source's more conservative radius 779: failure implies $|\Delta| \geq 780$. No Gaussian approximation is used.

**Exact product moment-generating function** For a fixed label, expand the negacyclic coefficient and group the two products sharing each $r_{\ell,j}$, and the two products sharing each $(e_1)_{\ell,j}$. Up to signs, the result is the sum of **1,536 independent products** of a CBD2 variable with an independent CBD4 variable, plus an independent CBD4 variable.

The CBD4 variables are the sums or differences of the corresponding $e, f$ coefficients and $s, x$ coefficients. For a fixed label these are independent across $(\ell, j)$, because the index maps are permutations and the underlying four secret/noise arrays are independent. This grouping handles the shared $r$ and $e_1$ correctly; treating the original four inner products as independent would not.

For $B \sim \mathrm{CBD2}$, $\Pr[B = 0] = 3/8$, $\Pr[|B| = 1] = 1/2$, $\Pr[|B| = 2] = 1/8$. Since $\mathbb{E}e^{t\,\mathrm{CBD4}} = \cosh(t/2)^8$, one product has MGF

$$M(t) = \frac{3}{8} + \frac{1}{2}\cosh(t/2)^8 + \frac{1}{8}\cosh(t)^8.$$

For $t > 0$, the following bound holds:

$$\Pr[|\Delta_{\tau_0,\epsilon_0,m}| \geq 780] \leq 2e^{-780t}M(t)^{1536} \cdot \cosh(t/2)^8.$$

There are $2n^2 = 131,072$ possible labels. Union-bound **all** of them before applying the actual selector. The resulting event controls every selected occurrence, even if selection is correlated with the entire noisy ciphertext:

$$\delta_{\mathrm{rec}} \leq 2^{18}e^{-780t}M(t)^{1536}\cosh(t/2)^8.$$

Taking the fixed rational value $t = 6/25$ gives

$$\boxed{\delta_{\mathrm{rec}} < 3 \cdot 2^{-121}.}$$

The raw expression is approximately $2^{-119.4867}$; the boxed rational bound is the certified value used below. This is a conservative upper bound, not an estimate of the true error rate or an optimized security-level claim.

**Structural rejection: 65 zero CRT factors** The ring has 128 quadratic factors over $\mathbb{F}_q$. Fewer than 64 active factors means at least 65 factors on which all three components of $u$ are zero. The all-zero $u$ case is included.

For any fixed set of $z$ factors, simultaneous vanishing of one polynomial imposes $2z$ independent linear constraints on its 256 coefficients. Choose pivot coefficients. After conditioning on all others, each pivot CBD2 coefficient has point probability at most $3/8$, so a three-polynomial vector vanishes there with probability at most $(3/8)^{6z}$.

Fix a subset $\mathcal{I}_{\mathrm{CRT}}$ of 65 factors and partition according to $Z \subseteq \mathcal{I}_{\mathrm{CRT}}$, the factors where $r$ vanishes. At these factors $u = e_1$, giving a joint bound $(3/8)^{12|Z|}$ for $r$ and $e_1$. At every other factor, the nonzero vector $r$ makes $A^T r$ uniform in the three-component quadratic-field vector, independently across those factors, so the probability of zero $u$ is $q^{-6}$ per factor. Summing over $Z$ and then over $\mathcal{I}_{\mathrm{CRT}}$,

$$\begin{aligned}\delta_{\mathrm{adm}} &\leq \binom{128}{65}\left[(3/8)^{12} + q^{-6}\right]^{65} \\ &\leq 2^{128}\left[(3/8)^{12} + q^{-6}\right]^{65} \\ &< 2^{-975}.\end{aligned}$$

Dropping the exact-zero-pattern constraints only increases the bound. This argument does not assume independent CRT evaluations of a CBD polynomial.

**The actual matrix-expansion cap** Each matrix entry parses the first 1,024 XOF bytes into 682 independent 12-bit candidates, accepting values below 3,329. The SHAKE128 rate is 168, divisible by three; the final partial block contributes ten candidates, so the source's block parsing gives the same 682 candidates. Let $p_{\mathrm{samp}} = 3329/4096$ and $Y \sim \mathrm{Bin}(682, p_{\mathrm{samp}})$. A zero-filled tail can occur only if $Y \le 255$.

Using the exponential Markov inequality at parameter 1, then union-bounding nine matrix entries,

$$\delta_A \le 9e^{255}(1 - p_{\mathrm{samp}} + p_{\mathrm{samp}}e^{-1})^{682} < 2^{-338}.$$

Couple to an infinite-rejection uniform matrix until this event. The bounds on decoding and the active filter can therefore be combined with this additional term. Canonical encoding follows from exact modular arithmetic in the source, so there is no separate random canonicality loss.

**Ideal-oracle coupling and prior history** Use the same independent ideal interfaces for SHA3-512, SHAKE128 and SHAKE256 that instantiate the sampling experiment. With no pre-key-generation oracle activity, the key-generation noise inputs $\sigma\|0, \ldots, \sigma\|3$ are distinct, fresh 33-byte inputs. Their streams give independent CBD2 coefficients.

After public catalog preprocessing and before the honest challenge, let $B_{33}$ bound the number of distinct 32-byte prefixes of already queried 33-byte SHAKE256 inputs. The fresh encapsulation coins $\omega$ are uniform and independent of that history. Thus

$$\Pr[\omega \text{ equals one of these prefixes}] \le B_{33}/2^{256}.$$

Outside this event, the four encapsulation noise streams $\omega\|0, \ldots, \omega\|3$ are fresh and independent of the key-generation arrays and the preceding transcript. A game that replaces a repeated-prefix encapsulation by fresh independent streams differs only on the displayed event. This permits the unconditional ideal-CBD calculation; one must not simply assert independence after conditioning on an arbitrary observed transcript.

Before the honest challenge, at most 70,656 catalog challenge-seed prefixes and 6,144 standard oracle-Decaps challenge-seed prefixes can occur among SHAKE256 inputs of length 33 bytes; the four key-generation noise streams share one additional prefix. Forest inputs have length 1,156. Therefore $B_{33} \le 70,656 + 6,144 + 1 = 76,801$, counting adaptive queries conservatively, and

$$\delta_{\mathrm{seed}} \le 76,801/2^{256} < 2^{-239}.$$

The complete honest-correctness estimate is

$$\delta_{\mathrm{cor}} \le 3 \cdot 2^{-121} + 2^{-975} + 2^{-338} + 2^{-239} < 2^{-119}.$$

This is an average-over-KeyGen/Encaps statement in the specified model. For an arbitrary larger prior history, keep $B_{33}/2^{256}$ explicit. For the fixed real sampler, the rigorous transfer statement is the ideal bound plus the difference in this efficiently recognizable correctness event between real and ideal sampling. Collision resistance of SHAKE alone does not prove that distributional transfer.

## 4.6 Experimental Evaluation

The preceding sections establish the source-specific extraction mechanism and its ideal-model bounds. We now compare the three executable query schedules on the same recovered states. We evaluated the published reference C routines, public probe catalog, signed-graph solver and local key predictor in separately compiled processes. The extraction client received only ordinary 32-byte Decaps outputs; a different verifier compared its recovered 1,536 signed coordinates against KeyGen's original state. Predictions are computed solely from recovered state; the verifier's ground-truth coefficients are not supplied to the client.

The baseline selects one of 184 precomputed *adaptive binary decision trees*, indexed by 92 admissible gains $b \in [104, 195]$ and two orientations. Source parsing verifies that every tree has one depth-one leaf for an extreme value $v = \pm 4$ ($\mathrm{CBD}_4$ mass $1/256$) and eight depth-four leaves. The weighted mean is therefore $4 - 3/256 = 3.98828125$ comparisons per relation, or 6,126 per 1,536 relations under the nominal $\mathrm{CBD}_4$ prior. The optimized schedule minimizes the per-scalar weighted path length among ordered depth-four trees.

**Experimental design** The evaluation used 50 independently generated OS-CSPRNG keys in five ten-key batches on an Ubuntu x86-64 research host. For each key, an isolated owner regenerated the same key for the original-tree, depth-four scalar, and graph-aware schedules. All three clients used the same pinned reference routines and observed only ordinary 32-byte Decaps outputs; no KeyGen coins or secret coefficients were provided to the clients. Each batch produced 90 raw JSONL records, for 450 records in total. Public-key hashes and seed commitments agree across schedules for each key, while the 50 public-key hashes are distinct. Details of the build, process isolation and supporting implementation checks are retained in Appendix A and the reproducibility artifact.

**Table 6:** Three query schedules evaluated on the same 50 matched keys. CBD model values are nominal prior expectations conditional on graph-basis selection.

| Schedule | Calls/key | CBD model | Keys |
|---|---|---|---|
| Original trees | 6,127.38 | 6,126 | 50 paired |
| $\mathrm{CBD}_4$ depth-four | 4,382.54 | 4,380 | same 50 |
| Graph-aware peeling | 3,565.18 | 3,565.56 | same 50 |

All 50 matched per-key call counts and cycle sizes are reported in Table 11 of Appendix A; the raw records are retained in the reproducibility artifact.

In all 150/150 compiled recoveries, the attack reconstructed the 1,536 secret coefficients, giving 230,400 exact coordinate comparisons. The 50 KeyGen instances provided 20 honest challenge events per key, for 1,000 distinct honest challenges and 3,000/3,000 successful key predictions across the three recovered states. None of the 3,000 negative-control comparisons produced a false match. Each complete recovery used fewer than 6,144 ordinary Decaps calls.

**Observed query comparisons** For the same 50 keys, the original schedule averaged 6,127.38 ordinary decapsulation calls, the depth-four scalar schedule 4,382.54, and graph-aware peeling 3,565.18. Graph-aware peeling therefore saved 817.36 calls per key (18.65%) relative to the executable depth-four scalar baseline, and 41.82% relative to the original schedule. All reductions count online decapsulation calls, not elapsed time or offline preprocessing.

Graph-aware peeling used $C_{\mathrm{cyc}} \in [8, 72]$ cycle edges (mean 25.16). Every graph-aware recovery completed in 3,525–3,602 standard Decaps calls. The nominal CBD-marginal model predicts 3,565.5628125 calls at the observed mean cycle count, compared with the measured 3,565.18. For the largest observed 72-cycle-edge basis, the certified worst-case

query bound is $4,608 + 72 = 4,680$. Agreement of the means does not itself prove that public-key-dependent basis selection preserves independent CBD marginals.

Post-hoc censoring at $Q_{\max} = 4,500$ yields completion counts of 0/50, 50/50, and 50/50 for the original, depth-four scalar, and graph-aware schedules, respectively. At $Q_{\max} = 4,000$, the counts are 0/50, 0/50, and 50/50. These budgets were applied to completed traces; no rate limiter or key rotation was enforced during the experiment.

**Paired-key statistical uncertainty** The depth-four scalar and graph-aware schedules were evaluated on exactly the same 50 keys. Their mean paired saving was 817.36 online decapsulation calls (sample standard deviation 24.19). The Student $t$ critical value for 49 degrees of freedom gives an approximate 95% paired interval of 810.49–824.23 calls. Savings were positive on all 50 keys (observed range 757–865 calls). The interval assumes independently sampled keys and approximately normal paired differences; it is not a distribution-wide guarantee.

### 4.7 Auxiliary Scalar Amplitude Frontiers

The measured improvement relies on the fixed source-admitted gain range; it does not require the more general amplitude bounds developed below. The source-admitted attack uses $104 \le b \le 195$ to isolate one verifier bit at a fixed public occurrence. Related scalar-access models allow different gain choices for the same nine-valued relation and have sharp minimum amplitudes 92, 53, and 52 for fixed, adaptive positive, and signed gains, respectively. These are auxiliary decoder results, not alternative measured HCR-KEM attacks: changing the monomial core can change the public challenge selector. The precise observation models, statements, and proofs appear in Appendix A.3.1.

## 5 Design Lessons, Countermeasures and Related Work

The attack raises two related questions: when can a recovered coefficient make subsequent observations cheaper, and which validation rules prevent such observations in the first place? We address the first through a conditional criterion and the second through the returned-key behavior of a standard KEM.

### 5.1 Reconciliation Independence Criterion

Separate information gained from the recovery transcript from any additional query cuts enabled by a known neighboring coefficient. For a fixed posterior $\pi$ on unresolved coordinates, let $\mathcal{Q}_0(\pi)$ denote the admissible binary observations without using a recovered neighbor, and $\mathcal{Q}_k(\pi)$ those available when the neighbor has known value $k$. Queries are *cut-equivalent* if they induce the same response distribution on every remaining state with the same query cost.

**Proposition 2** (Reconciliation Independence Criterion)**.** *Consider a finite noiseless binary recovery problem. Suppose that at each reachable residual posterior $\pi$ and for every admissible known neighbor value $k$, the nontrivial cut-equivalence classes of $\mathcal{Q}_k(\pi)$ coincide with those of $\mathcal{Q}_0(\pi)$, including their costs and successor posterior distributions. Then the optimal expected number of* remaining *queries is the same for these two models at the same posterior $\pi$. Knowing the neighbor cannot yield a strict reduction solely by enlarging the realizable cut family.*

*Proof.* For a point-mass posterior both costs vanish. For larger finite support, consider any informative cut. The assumption supplies an equivalent cut of identical cost in the other model. Each nonempty successor has strictly smaller support, so induction gives identical

continuation costs. The two Bellman objectives therefore coincide for every corresponding cut. Minimizing over the same equivalence classes proves the claim. □

This is a sufficient condition for absence of a *cut-family* peeling gain, not an IND-CCA theorem. Its converse is false: additional cuts need not lower the optimum. Earlier queries that recover the neighbor are not free, and adaptive conditioning may change $\pi$. A concrete implementation must establish the cut families from the exact decryption, reconciliation, ciphertext-format and validation algorithms.

**Conditional observable-interface diagnostic** For a reused secret state $S$, an adaptive pre-challenge algorithm may form $(c_i, k_i)$ and observe $e_i = [\mathsf{Decaps}(sk, c_i) = k_i]$. The observation supports extraction when (i) each $e_i$ predicts a nonconstant predicate of $S$ with conditional error at most $\epsilon_i$ uniformly over all reachable prior transcripts, (ii) the resulting constraints admit independent checking, and (iii) a deterministic solver uniquely reconstructs the state or incurs bounded structural failure $\delta_{\text{struct}}$. With a fixed query limit $Q$,

$$\Pr[\text{Ext fails}] \le \delta_{\text{struct}} + \sum_{i=1}^{Q} \epsilon_i.$$

## 5.2 A source-bound negative case: standard ML-KEM rejection

The following boundary follows directly from the returned-key formula on the guaranteed-rejection branch; it does not require the cut-equivalence hypothesis of Proposition 2. The difference between confirmation-only validation and complete ciphertext consistency can be made precise on a published interface. Algorithm 18 of NIST FIPS 203 [19] derives the candidate key $K' = G_1(m' \| H(pk))$, re-encrypts the decoded message to obtain $c'$, and replaces the candidate key by $\overline{K} = J(z\|c)$ if $c' \neq c$. The internal rejection flag is not returned.

**Proposition 3** (Guaranteed-rejection transfer boundary)**.** *Fix a public encapsulation key, implicit-rejection seed $z$, and a posterior set $\mathcal{S}$ of possible K-PKE decryption states. Restrict the source-admissible queries to ciphertexts $c$ for which the FIPS 203 re-encryption test fails for* every *state $s \in \mathcal{S}$. For any publicly chosen candidate key $k(c)$, the returned-key equality observation $[\mathsf{Decaps}(s, z, c) = k(c)]$ is constant over $\mathcal{S}$. Consequently this restricted query family induces no nontrivial binary cut on the decryption-state posterior, before or after any other coefficient is revealed.*

*Proof.* By the rejection branch of Algorithm 18, the returned key on every considered ciphertext is $J(z\|c)$ for all $s \in \mathcal{S}$. The equality comparison is therefore $[J(z\|c) = k(c)]$, which has no dependence on $s$. Conditioning on one or more coordinates of $s$ does not change this cut family. □

This is an exact *rejection-branch* negative transfer for the standard ML-KEM interface, not a proof that every ciphertext is rejected, that accepted ciphertexts admit no informative cuts, or that the K-PKE decryption key cannot be recovered in stronger leakage or fault models. In particular, a plaintext-checking oracle, an exposed internal rejection flag, or faulty FO verification changes the available observations. The result shows that the HCR returned-key equality construction cannot be transferred by simply reusing guaranteed-rejected ciphertexts against correctly implemented ML-KEM.

## 5.3 Validation and construction lessons

The contrast between HCR-KEM's admitted probes and ML-KEM's guaranteed-rejection branch points to the central design lesson: the security question is what an adversary can

learn from the complete returned-key interface after ciphertext validation. Confirmation of a decoded verifier word does not, by itself, authenticate every component of the ciphertext that produced it. The HCR example exhibits source-admitted probes for which a predicted confirmation word leaves a secret-dependent returned-key equality predicate. A properly instantiated FO re-encryption check is a separate form of ciphertext consistency validation [19, 20]. It is not equivalent to hashing the recovered word. Thus, a lightweight activity filter and a matching confirmation tag should not be treated as substitutes for validating the ciphertext's full consistency. We do not claim to have implemented or proved an IND-CCA-secure patched HCR variant; a redesigned decoupling transform would need an independent correctness and security proof.

**FO-style redesign obligation** This lesson suggests a concrete redesign requirement rather than a change to the existing proof: validation must bind the accepted key to the complete ciphertext. A possible redesign would use a message-bearing, deterministic encapsulation component with explicit recoverable plaintext $m$, derive encapsulation coins and a candidate shared key from $m$ and the public key, and require decapsulation to reconstruct $c' = \mathsf{EncDet}(pk, m')$ and compare it with the entire received ciphertext $c$. A mismatch would select a domain-separated implicit-rejection key $J(z\|c)$; a match would select the candidate key. The comparison and key selection must not expose a validity signal. This describes the consistency mechanism in FIPS 203 [19], not a tested HCR patch. HCR's present source accepts coins directly in encapsulation and does not implement the required message-bearing PKE interface. Consequently one cannot simply append a ciphertext comparison and claim FO security: the message encoding, correctness/failure probability, deterministic re-encapsulation, domain separation, admissible ciphertext validation, and IND-CCA reduction must each be redesigned and proved.

## 5.4 Prior work and comparison boundaries

The main result is a source-admitted, tag-gated returned-key equality oracle that survives the activity filter and isolates a single signed relation despite pseudorandom forest selection. Its optimal threshold-query schedule and publicly checkable graph certificate turn that observable into complete state recovery. Previous work supplies the foundations of mismatch checking and boundary search [2–4], signed and parallel coefficient cuts [5–7], multiple-valued observations [10], signed-incidence rank [11], and general distribution hints [12]. The scalar-amplitude lower bounds of Section 4.7 concern restricted observation models and are auxiliary to the source-based attack.

**Additional recent mismatch-attack baselines** The multi-bit inference and ciphertext generalization of Shao et al. [8] and the Huffman-optimal ML-KEM analysis [9] provide essential comparators when assessing novelty beyond HCR-KEM. These papers use their own specified mismatch, key-reuse or side-channel settings; online-query counts cannot be compared across different oracle outputs, offline workloads or recovery objects. We make no cross-scheme query reduction claim.

Table 7 compares the observable interfaces and recovered objects underlying these attacks rather than ranking unlike query counts. The scalar-amplitude frontier and signed-graph certificate address complementary questions.

**Relationship to chosen-ciphertext reconciliation and key-mismatch attacks** Key mismatch and reaction observations have previously led to recovery of reused secrets in lattice-based constructions. In particular, Bernstein et al. demonstrated chosen-ciphertext secret-key recovery against the submitted HILA5 *KEM*, not only against unauthenticated key agreement [3]. Bauer et al. analysed the key-reuse resilience of NewHope [13]. Its

**Table 7:** Closest precedents by observable interface, recovered object, and reconstruction method.

| Work | Observable interface | Recovered object | Main method |
|---|---|---|---|
| Multi-positional mismatch [5, 6, 8, 9] | CPA-PKE / PC-oracle leakage under reuse | Kyber and ML-KEM secrets | Pairwise, parallel and Huffman-based schedules |
| HILA5 [3] | Predicted returned KEM key | Reused KEM secret | Adaptive boundary search |
| Limited key reuse [7] | Mismatch under reuse | Kyber secret | Adaptive relations, lattice postprocessing |
| NewHope [13] | CPA-KEM reuse mismatch; CCA needs leakage/fault | Reused secret information | Encryption-based KEM analysis |
| MAMBA-NIKE [14] | NIKE reconciliation / key outputs | Static NIKE secret | Interface-dependent recovery |
| Faulty FO verification [1] | Incomplete rejection check | ML-KEM secret | Unverified-coordinate exposure |
| This HCR implementation | Ordinary 32-byte Decaps key | Full $(s, x)$: 1,536 recovered coefficients | Tag-gated bit, graph certificate |

Section 3 distinguishes direct CPA-KEM key-mismatch observations under reuse from CCA-KEM observations requiring a fault or side channel. NewHope is studied there as an encryption-based KEM, not itself a generic reconciliation KEM, and neither model is equivalent to ordinary access to a fully verified decapsulation interface. More recently, Xiong and Wang studied static-key reconciliation in MAMBA-NIKE [14]. Their public raw-reconciliation experiments use a `-DSTATISTICAL_TEST` build to expose intermediate bits, and the independent candidate report describes complete static-key recovery in 71 such queries for its 128- and 384-bit profiles. It does not establish an equivalent 71-query attack on the usual hashed session-key interface. The independently maintained `ngcc.dev` report [15] describes raw-reconciliation recovery for its stated parameter sets and gives an estimated, unbenchmarked hashed-key reaction route. This site is independent of the ICCS/NGCC organizers. MAMBA-NIKE is a real candidate with an unauthenticated static-key reconciliation path, so it illustrates the relevance of scrutinizing the observer's exact interface. It is *not* evidence that the source admits HCR-KEM's particular 290-bit confirmation-tag forgery or our unit-monomial forest-isolation construction. These works establish that the attacker's observable, the treatment of reused secrets and the strength of the interface matter to the attack's complexity. Together, they motivate precise statements of the oracle and key-reuse conditions under which a complete secret-recovery result is obtained.

**Relationship to FO confirmation and implicit rejection** Hövelmanns et al. formalize computational rigidity and the role of re-encryption or alternative validation in the Fujisaki–Okamoto transform [16]. Hövelmanns and Kudinov examine explicit and implicit rejection as distinct FO mechanisms [17]. Glabush et al. study *verifiable decapsulation*, deliberately making implementation omissions of verification observable through correctness tests [18]. The HCR-KEM observation studied here is different: the adversary locally compares the *returned shared key* from a standard decapsulation query with a public candidate key. It does not assume a separate validity bit, an implementation fault or a physical side channel. Conversely, this observation is not, by itself, a generic bypass of a correctly implemented FO transform. The HCR argument requires a source-admitted ciphertext construction for which this comparison reveals a *nontrivial* predicate of an unknown reused secret.

**Distinctness from faulty FO rejection attacks** Das [1] studies ML-KEM/Kyber implementations where a Fujisaki–Okamoto rejection operation is skipped or a ciphertext-comparison coordinate is left unchecked. Such defects can expose an oracle sufficient for secret-key recovery. This is a different attack premise from the present HCR returned-key comparison: no defective rejection check or separate validity bit is assumed here. Conversely, the HCR evidence does not establish a flaw in correctly implemented ML-KEM. Complete re-encryption comparison and implicit rejection are relevant design controls, but eliminating the specific HCR equality predicate requires a separate, variant-specific proof.

**Application to reconciliation and NIKE interfaces** The same diagnostic motivates scrutiny of reused-secret reconciliation interfaces, including NewHope-style helper data and MAMBA-NIKE [13–15]. A protocol designer should determine which peer inputs can be chosen under a reused secret, whether the helper, confirmation tag, key or error signal is externally observable or locally predictable, whether validation covers the complete accepted ciphertext or transcript, and whether admissible changes can isolate a nonconstant secret-dependent predicate. This checklist concerns the actual interface: the reported MAMBA-NIKE raw-reconciliation experiment does not by itself establish the same oracle on its normal hashed session-key output.

### 5.5 Scope and Limitations

The result concerns the publicly specified HCR-KEM research construction and its ordinary returned-key interface, not a standardized or deployed KEM. The reference C experiments support exact recovery and matched-key query counts for the tested builds; they are distinct from a formal machine-code proof. Finite arithmetic validation and compiler checks are documented in Appendix A and the artifact.

The probability statements assume exact ring arithmetic and the restricted-history ideal-oracle experiment, including the stated fresh-stream and collision accounting. The independently reproduced numerical expressions do not by themselves verify every source-to-oracle or conditional-independence premise. Similarly, the matched-key confidence interval describes the measured paired sample, not a distribution-wide guarantee. Nominal CBD-marginal cost estimates require the stated assumptions on public-key-dependent basis selection.

The experimental comparison uses matched keys and a single query-counting protocol; its results are not extrapolated to other constructions. The reproducibility records preserve source identities and public commitments, but independent source-to-compiled replay and verification of the ideal-oracle assumptions remain distinct obligations. Internal reconciliation properties in other schemes establish neither an attacker-admissible observation nor a new HCR-KEM key recovery.

## 6 Conclusion and Future Work

We presented HCR-KEM as a fully specified reconciliation-based research KEM and analyzed the security of its confirmation-only decapsulation interface. By combining an admitted unit monomial with a locally computed whole-word confirmation tag, we isolate one signed relation while 289 other forest positions remain fixed. The ordinary 32-byte returned key provides the equality observation, and a public signed-graph certificate determines the full 1,536-coefficient secret. Our exact-arithmetic analysis also gives explicit catalog availability and honest-correctness bounds under the stated ideal-model assumptions.

The recovery mechanism also explains the main algorithmic contribution, graph-aware peeling. Instead of treating every signed relation as an independent nine-valued

threshold problem, we first resolve the cycle-closing relations and then use recovered parent coefficients to identify the remaining tree vertices through five-valued searches. In the 50-key matched evaluation, all 150 recoveries succeeded. Graph-aware peeling reduced mean online decapsulation calls from 4,382.54 under the depth-four scalar baseline to 3,565.18, a paired saving of 817.36 calls (18.65%). The measured improvement follows from conditioning the query schedule on the signed-graph reconstruction, rather than from a different solution to the same independent scalar problem.

The broader lesson is that confirming a reconciled word is not the same as validating the ciphertext that generated it. In HCR-KEM, the admitted probes leave enough of that word predictable to turn an ordinary returned-key comparison into a secret-dependent observation; in contrast, our ML-KEM analysis identifies a precise boundary on the guaranteed-rejection branch, where a fixed rejection seed makes the returned key independent of the K-PKE secret coefficients. This distinction makes admissible ciphertexts, complete returned-key behavior, and the availability of conditional query cuts essential parts of a security assessment. Future work will examine these properties in published reconciliation and KEM implementations, including NewHope and the CPA cores of Kyber/ML-KEM and Saber, before testing whether graph-aware peeling transfers. Accepted-ciphertext observations and stronger oracle models require separate analysis.

## Ethics Considerations

We tested a public reference-C implementation with ephemeral keys in isolation, without accessing deployed services or private data. The artifact contains source and aggregate results, not secret keys or credentials. We make no vulnerability claim about any other works.

## Reproducibility and Data Availability

The artifact contains pinned C sources, catalog, three recovery clients, isolated owner and verifier, local predictor, graph-cycle accounting, raw receipts and replay scripts, with SHA-256 identities in its manifest. The 50 matched keys were drawn using an OS-CSPRNG; their public-key hashes and seed commitments are recorded, while the erased random coins cannot be reconstructed from commitments. The procedure can be repeated with independently generated keys, not with the identical erased random draws. The reproduction package accompanying this revision includes the archived source and measurement records. A stable hosting URL will soon be available here.

**Independent replay criteria** Independent replay should regenerate 70,656 catalog rows, verify signed-graph closure, recover secrets using only ordinary decapsulation outputs and check honest-key predictions from recovered coefficients. Ideal-oracle premises are distinct from finite C tests.

## References

[1] Bhabani Sankar Das, *One Unverified Coordinate Is Enough: Key Recovery from Faulty Fujisaki–Okamoto Rejection Checks in ML-KEM*, Cryptology ePrint Archive, Paper 2026/2239, 2026. https://eprint.iacr.org/2026/2239.

[2] Jintai Ding, Scott Fluhrer, and Saraswathy Rv. *Complete Attack on RLWE Key Exchange with Reused Keys, Without Signal Leakage*. ACISP 2018, pp. 467–486. https://doi.org/10.1007/978-3-319-93638-3_27. Primary text: https://archive.ymsc.tsinghua.edu.cn/pacm_download/672/12661-dingjt-c63.pdf.

[3] Daniel J. Bernstein, Leon Groot Bruinderink, Tanja Lange, and Lorenz Panny. *"HILA5 Pindakaas": On the CCA Security of Lattice-Based Encryption with Error Correction*. ePrint 2017/1214; AFRICACRYPT 2018. https://pure.tue.nl/ws/portalfiles/portal/92531109/bernHILA2017.pdf.

[4] Yue Qin, Chi Cheng, Xiaohan Zhang, Yanbin Pan, Lei Hu, and Jintai Ding. *A Systematic Approach and Analysis of Key Mismatch Attacks on Lattice-Based NIST Candidate KEMs*. ASIACRYPT 2021; ePrint 2021/123. https://eprint.iacr.org/2021/123. https://doi.org/10.1007/978-3-030-92068-5_4.

[5] Qian Guo and Erik Mårtensson. *Do Not Bound to a Single Position: Near-Optimal Multi-Positional Mismatch Attacks Against Kyber and Saber*. PQCrypto 2023, pp. 291–320; ePrint 2022/983. https://eprint.iacr.org/2022/983.

[6] Mingyao Shao, Yuejun Liu, and Yongbin Zhou. *Pairwise and Parallel: Enhancing the Key Mismatch Attacks on Kyber and Beyond*. ePrint 2023/887; AsiaCCS 2024. https://eprint.iacr.org/2023/887. https://doi.org/10.1145/3634737.3637661.

[7] Qian Guo, Erik Mårtensson, and Adrian Åström. *The Perils of Limited Key Reuse: Adaptive and Parallel Mismatch Attacks with Post-processing Against Kyber*. IACR Communications in Cryptology 1(3), 2024. https://doi.org/10.62056/a3n5qj888. Full text: https://cic.iacr.org/p/1/3/21/pdf.

[8] Yan Shao, Yuejun Liu, Yongbin Zhou, and Mingyao Shao. *Enhanced Key Mismatch Attacks on Lattice-Based KEMs: Multi-bit Inference and Ciphertext Generalization*. ESORICS 2025, Part II, pp. 341–358. https://doi.org/10.1007/978-3-032-07891-9_18.

[9] Mingyao Shao, Yuejun Liu, Yongbin Zhou, Rui Zhang, Yanbin Pan, Zehua Qiao, Yan Shao, and Xingyang Huang. *Closing the Gap: Huffman-Optimal and Stealthy Key-Mismatch Attacks on ML-KEM*. ESORICS 2026, Part II, pp. 341–362. https://doi.org/10.1007/978-3-032-38695-3_17.

[10] Yutaro Tanaka, Rei Ueno, Keita Xagawa, Akira Ito, Junko Takahashi, and Naofumi Homma. *Multiple-Valued Plaintext-Checking Side-Channel Attacks on Post-Quantum KEMs*. TCHES 2023(3), pp. 473–503. https://doi.org/10.46586/tches.v2023.i3.473-503. Primary paper text: https://d-nb.info/1307299806/34.

[11] Thomas Zaslavsky. *Signed Graphs and Geometry*. IWSSG 2011 lecture notes, Theorems 3.5–3.6. https://people.math.binghamton.edu/zaslav/Tpapers/mananthavady-lecture-notes.20110729.pdf.

[12] Julius Hermelink, Silvan Streit, Erik Mårtensson, and Richard Petri. *A Generic Framework for Side-Channel Attacks against LWE-based Cryptosystems*. EUROCRYPT 2025; revised ePrint 2024/1211. https://eprint.iacr.org/2024/1211.

[13] Aurélie Bauer, Henri Gilbert, Guénaël Renault, and Mélissa Rossi. *Assessment of the Key-Reuse Resilience of NewHope*. CT-RSA 2019, LNCS 11405, pp. 272–292. https://doi.org/10.1007/978-3-030-12612-4_14. IACR ePrint 2019/075, https://eprint.iacr.org/2019/075.

[14] Zhenyu Xiong and Mingsheng Wang. *Cryptanalysis of the ICCS NGCC Round-1 Public-Key Candidates*. IACR ePrint 2026/2232 (2026). https://eprint.iacr.org/2026/2232.

[15] *ngcc.dev*, *MAMBA-NIKE (kex-06)*, independent security tracker maintained by M.-J. O. Saarinen. Finding kex-06-2 credits Z. Xiong and M. Wang (accessed 17 November 2026). https://ngcc.dev/reports/kex-06.html.

[16] Kathrin Hövelmanns, Andreas Hülsing, Christian Majenz, and Fabrizio Sisinni. *(Un)breakable Curses: Re-encryption in the Fujisaki–Okamoto Transform*. EUROCRYPT 2025, LNCS 15602, pp. 245–274. https://doi.org/10.1007/978-3-031-91124-8_9. IACR ePrint 2025/299, https://eprint.iacr.org/2025/299.

[17] Kathrin Hövelmanns and Mikhail Kudinov. *Treating Dishonest Ciphertexts in Post-Quantum KEMs: Explicit vs. Implicit Rejection in the FO Transform*. IACR ePrint 2025/062. https://eprint.iacr.org/2025/062.

[18] Lewis Glabush, Felix Günther, Kathrin Hövelmanns, and Douglas Stebila. *Verifiable Decapsulation: Recognizing Faulty Implementations of Post-Quantum KEMs*. CRYPTO 2025, LNCS 16002, pp. 543–574. https://doi.org/10.1007/978-3-032-01881-6_17. IACR ePrint 2025/450, https://eprint.iacr.org/2025/450.

[19] National Institute of Standards and Technology. *Module-Lattice-Based Key-Encapsulation Mechanism Standard*. FIPS 203, Aug. 2024, Algorithm 18. https://doi.org/10.6028/NIST.FIPS.203.

[20] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. *A Modular Analysis of the Fujisaki–Okamoto Transformation*. TCC 2017, pp. 341–371; IACR ePrint 2017/604. https://doi.org/10.1007/978-3-319-70500-2_12.

[21] Korean Post-Quantum Cryptography Standardization Project (KpqC), *SMAUG-T: Final Specification*, 2026, Fig. 8 (decapsulation), p. 19. https://www.kpqc.or.kr/images/pdf2/SMAUG-T.pdf.

[22] Korean Post-Quantum Cryptography Standardization Project (KpqC), *NTRU+: Final Specification*, Jan. 30, 2026, pp. 2, 6 (randomness-recovery validation). https://www.kpqc.or.kr/images/pdf2/NTRU%2B.pdf.

# A Source binding and reproducible analytical checks

The artifact records source commitments, aggregate public-sandbox results, and the scope and unresolved premises of the conditional theorems. Its provenance records do not substitute for proof or for independent review of this manuscript.

The separately supplied artifact identifies the reference C translation units, headers, pinned hashes, build configurations, independent verifiers, public-catalog receipts and the evaluation records. The normal production selector uses the challenge digest specified in Section 4.1; the diagnostic helper with an additional domain byte is not used to define the attack observation. The finite compiled tests and source-replay instructions are provided with the artifact. The mathematical results separately require the specified exact ring arithmetic, uniformly supplied coins and restricted-history ideal-oracle assumptions. None of these premises follows from the artifact hashes alone.

## A.1 Forest initialization and retry-zero finalizations

The proof of Lemma 3 uses the fact that forest components only merge: a candidate pair whose endpoints remained disconnected at the end of a complete scan must have been accepted when encountered.

*Proof.* At shift index $j$, after $r \le 9$ acceptances, the current forest has $E = 10j + r$ edges. Let $P$ of the 256 pairs in the current shift's perfect matching already have connected endpoints. Distinct shifts have disjoint edge sets. In each tree component, if $p$ such pairs are internal and $r_c$ of its edges belong to the current matching, then $3p \le 2e + r_c$, where $e$ is its edge count. Indeed, $p \le \lfloor v/2 \rfloor$ for $v$ vertices. For $v \ge 3$, $3\lfloor v/2 \rfloor \le 2(v-1)$; for $v = 2$, an internal current pair requires the component's single edge to be in that matching, giving $3 \le 2 + 1$. Isolated vertices contribute zero. Summing gives

$$3P \le 2E + r = 20j + 3r \le 587.$$

The last inequality permits equality at $j = 28$ and $r = 9$. Thus $P \le 195$, leaving at least $256 - 195 = 61$ disconnected pairs. If the first pass ended with fewer than ten acceptances, it would have visited every present stream byte and every absent byte, hence all 256 candidates. Any pair disconnected at the end was disconnected when visited, because components only merge; it would have been accepted, a contradiction. Induction over the shifts proves the claim. The raw loop's upper bound $N_{\text{forest}} \le 232$ remains valid for arbitrary caller-supplied shifts, for which this distinct-shift argument need not apply. □

**First-shift initialization** The first shift begins from 512 singleton vertices. Its first XOF byte $m \in \{0, \ldots, 255\}$ selects the edge from left vertex $m$ to right vertex $256 + ((m - \tau_0) \bmod 256)$, which must be accepted. The lemma's complete retry-zero scan then initializes all 290 coordinates. Uniformity of the first byte is a separate ideal-XOF assumption, not a consequence of the union-find argument.

**Implementation checks** The 150 matched-key recoveries built 70,656 source-derived rows per key and full-rank signed-graph reconstruction certificates. The artifact also retains 80 supplementary compiled recoveries under five ten-key implementation configurations. Those records are not pooled with the principal 50-key paired analysis. An internal cross-check reconciled the 450 matched-key records with the per-key table and aggregate query statistics without mismatches. This establishes archive-to-manuscript consistency, not independent re-execution of the decapsulation binary. The compiled adapter completed ten fail-early null-state checks. Additional finite C validation included exhaustive Barrett reduction over 46,721 signed integers and 165,960 instrumented whole-vector assertions.

These tests support the measured C implementation behavior; the separate ideal probability theorem uses the exact arithmetic and oracle history defined in Section 4.4.

**Subsequent compiled quantizer validation** An additional source-contract audit checked 1,048,576 signed-input/hint combinations per executable in nine Windows and Ubuntu GCC/Clang configurations, totaling 9,437,184 comparisons with no mismatches against the independent modular-distance reference. This validates the tested quantizer behavior; it does not independently establish every source-to-graph, normal-return observation, or ideal-oracle sampling premise. Full compiler matrices and receipts belong to the reproducibility package.

## A.2 Independent replay materials

The accompanying reproducibility package contains the exact source identities, build records, public-catalog and signed-graph verification procedure, original and matched-key receipts, and replay instructions. The full referee checklist is supplied as `ARTIFACT_VERIFICATION_CHECKLIST.tex` in that package. The mathematical scalar frontiers are retained below.

## A.3 Supplementary Scalar Frontier Derivations

We establish sharp amplitude thresholds for identifying one $v \in \{-4, \dots, 4\}$ using the inspected scalar decoder: 92 for a fixed gain, 53 for adaptive positive gains, and 52 when signed gains are allowed. These are different access models from the full KEM: changing or negating the monomial core also changes the source's public challenge-selector outputs. For the implemented single-occurrence construction, the stronger spacing requirement $b \geq 2a = 104$ (with $q = 64a + 1 = 3329$) and the non-target condition $4b \leq 780$ select $b \in [104, 195]$. Thus the scalar frontiers locate an idealized design-space threshold, while the larger implemented window supports source-specific bit isolation without aligning multiple public cores. The definitions and proof follow.

### A.3.1 Scalar observation models

Let $a \geq 5$ be an integer, $q = 64a + 1$, and $h \in \{0, \dots, 15\}$. Define the two centers

$$c_0(h) = \left\lfloor \frac{(2h+1)q + 32}{64} \right\rfloor = (2h+1)a,$$
$$c_1(h) = \left\lfloor \frac{(2h+33)q + 32}{64} \right\rfloor = (2h+33)a + 1.$$

Write $d_q$ for circular distance modulo $q$. The decoder is

$$\mathrm{Rec}(w, h) = \begin{cases} 0, & d_q(w, c_0(h)) \leq d_q(w, c_1(h)), \\ 1, & d_q(w, c_0(h)) > d_q(w, c_1(h)). \end{cases}$$

In particular, distance ties are assigned to bit zero. This is the decoder in the inspected project source, not an assumption that all KEM decoders behave this way.

An unknown integer $v \in \{-4, -3, \dots, 4\}$ is observed through binary responses $\mathrm{Rec}(bv, h)$. The integer $b$ is a *gain*; its absolute value is the amplitude. All observations concern the same scalar relation $v$. The three access models are:

1. **One fixed gain:** choose one positive integer $b$, then vary only $h$. One fixed negative orientation is equivalent by reflecting $v$.
2. **Positive gains:** adaptively choose any integer $b$ with $1 \leq b \leq C$, and any hint $h$.

3. **Signed gains:** adaptively choose any nonzero integer $b$ with $|b| \le C$, and any hint $h$.

The objective is the smallest possible amplitude budget that identifies every value of $v$, with a finite number of permitted queries. Additional offsets, other secret relations, public-key equations, physical leakage and multi-bit answers are excluded from this scalar observation problem. No lower bound here purports to exclude attacks using those extra resources.

In a KEM application, these scalar queries must first be realized through admissible ciphertexts and an isolated verifier bit. The existing anchored equality mechanism can supply a binary answer under its stated acceptance and locality conditions [2, 3]. That mechanism is a premise of the application, not the novel object of this classification.

### A.3.2 Exact decoder geometry

**Lemma 5** (Decision arc and central thresholds)**.** *The zero decision set is the reduction modulo $q$ of*

$$[(2h-15)a, (2h+17)a].$$

*For integer lifts satisfying $|w| \le 16a$,*

$$\mathrm{Rec}(w,h) = 0 \quad \Longleftrightarrow \quad w \ge (2h-15)a. \tag{8}$$

*Every $w \in [-a, a-1]$ has the same full 16-hint signature as zero.*

*Proof.* The two arcs between the centers have lengths $32a+1$ and $32a$. On the first arc the last zero integer is $c_0 + 16a$; on the other arc the integer midpoint $c_0 - 16a$ is included by the tie rule. This gives the decision arc. Its smallest upper endpoint is $17a$, outside the central range. The largest upper endpoint in a preceding modular lift is $47a - q = -17a - 1$, also outside the range. Thus only the lower endpoint is active there. The two thresholds nearest zero are $-a$ and $a$, giving the full-signature interval $[-a, a-1]$. □

For $a = 52$, the zero-hint zero interval is $[-780, 884]$. The common central full-signature cell is $[-52, 51]$. These are different objects: the former concerns one hint, and the latter concerns all hints together.

### A.3.3 A sharp three-model amplitude theorem

**Theorem 4** (Exact amplitude frontier)**.** *For the decoder and nine-value alphabet above, with $a \ge 5$, the minimum identifying amplitudes are*

$$\boxed{\begin{aligned} C_{\mathrm{fixed}} &= \left\lceil \frac{7a+1}{4} \right\rceil, \\ C_{+} &= a+1, \qquad C_{\pm} = a. \end{aligned}}$$

*In the positive-gain model, the two gains $a-1$ and $a+1$ suffice. In the signed model, the gains $+a$ and $-a$ suffice. Each model admits exact identification using at most four binary queries once its required scalar accesses are available.*

Theorem 4 follows from the three gain-access cases below.

**One fixed gain** For a positive gain $b$ with $4b \le 7a$, all nine inputs $bv$ lie inside $[-7a, 7a]$. On this interval the full decoder signatures change at odd multiples of $a$. The interval meets only eight signature cells: its left endpoint begins the first cell, and its right endpoint begins the eighth. Nine different scalar values cannot then be distinguished. Therefore $4b \ge 7a+1$ is necessary.

For sufficiency, set $b_0 = \lceil (7a+1)/4 \rceil$. For $j = 1, 2, 3, 4$,

$$(2j-1)a < jb_0 \le 2ja < (2j+1)a.$$

The positive inputs $jb_0$ occupy four different positive cells; the negative inputs occupy four different negative cells; zero occupies the central cell. All nine signatures are distinct. The odd decision boundaries between consecutive inputs supply all eight separating cuts, so a four-level binary search is available.

**Several positive gains: the sharper comparator** If every positive gain satisfies $b \le a$, the cases $v = 0$ and $v = -1$ always produce decoder inputs in $[-a, 0]$, a subset of the common central signature cell. They have identical responses for every hint and every allowed gain. An adaptive procedure cannot distinguish them, by induction on its transcript. Thus $C_+ \ge a + 1$.

For sufficiency, use $b_- = a - 1$ and $b_+ = a + 1$. Because $4(a+1) \le 16a$, Lemma 5 applies to every queried value. For an odd integer $t$, choosing $h = (t+15)/2$ yields

$$\mathrm{Rec}(bv, h) = 0 \quad \Longleftrightarrow \quad v \ge \left\lceil \frac{ta}{b} \right\rceil.$$

The following choices give every cut $k = -3, -2, \ldots, 4$:

| $k$ | $-3$ | $-2$ | $-1$ | $0$ | $1$ | $2$ | $3$ | $4$ |
|---|---|---|---|---|---|---|---|---|
| $t$ | $-3$ | $-3$ | $-1$ | $-1$ | $1$ | $1$ | $3$ | $3$ |
| $b$ | $a-1$ | $a+1$ | $a-1$ | $a+1$ | $a+1$ | $a-1$ | $a+1$ | $a-1$ |

For example, $\lceil -3a/(a-1) \rceil = -3$ and $\lceil -3a/(a+1) \rceil = -2$. The other six identities follow in the same way from $a \ge 5$. Thus $C_+ = a + 1$, with a four-query identification tree.

**Signed gains** If $|b| \le a - 1$, the cases $v = 0$ and $v = 1$ produce inputs in $[-a+1, a-1]$ and have identical full signatures. Hence $C_\pm \ge a$.

For an odd cut $k$, gain $+a$ and hint $h = (k+15)/2$ give

$$\mathrm{Rec}(av, h) = 0 \quad \Longleftrightarrow \quad v \ge k.$$

For an even cut $k$, use gain $-a$ and $h = (16-k)/2$. Then

$$\mathrm{Rec}(-av, h) = 0 \quad \Longleftrightarrow \quad v \le k - 1.$$

In this case bit one means $v \ge k$. These choices realize all eight cuts, proving $C_\pm = a$. Nine candidates require at least four binary queries in the worst case when identifying that scalar without other information; a balanced threshold tree attains four. This elementary scalar lower bound is not a global key-recovery bound.

**Concrete values and threshold instructions** At $q = 3329$ the frontier is

$$(C_{\mathrm{fixed}}, C_+, C_\pm) = (92, 53, 52).$$

Table 8 lists a gain/hint pair $(b, h)$ for each cut. The response column specifies the decoder bit representing that cut being true. The signed true-bit column applies to the signed construction. The fixed-gain and positive-gain constructions use bit zero for every cut.

The gain improvement must therefore be described with its comparator. Signed access lowers the fixed-gain minimum from 92 to 52, but only lowers the optimal positive-gain budget from 53 to 52. Matching requirements differ between these access models, so neither comparison by itself establishes a practical end-to-end speedup.

**Table 8:** Amplitude cut instructions. Only the signed construction uses the signed true-bit column; fixed and positive gain models use bit zero.

| Cut $v \geq k$ | Fixed gain $(b, h)$ | Signed $(b, h)$ | Signed true bit | Positive gains $(b, h)$ |
|---|---|---|---|---|
| $-3$ | $(92, 4)$ | $(52, 6)$ | 0 | $(51, 6)$ |
| $-2$ | $(92, 5)$ | $(-52, 9)$ | 1 | $(53, 6)$ |
| $-1$ | $(92, 6)$ | $(52, 7)$ | 0 | $(51, 7)$ |
| 0 | $(92, 7)$ | $(-52, 8)$ | 1 | $(53, 7)$ |
| 1 | $(92, 8)$ | $(52, 8)$ | 0 | $(53, 8)$ |
| 2 | $(92, 9)$ | $(-52, 7)$ | 1 | $(51, 8)$ |
| 3 | $(92, 10)$ | $(52, 9)$ | 0 | $(53, 9)$ |
| 4 | $(92, 11)$ | $(-52, 6)$ | 1 | $(51, 9)$ |

### A.3.4 Why reflection gives extra information

**Proposition 4** (Reflection reveals the distance-tie case)**.** *For $h^* = 15 - h$, the centers satisfy*

$$c_0(h^*) = q - c_1(h), \qquad c_1(h^*) = q - c_0(h).$$

*For any residue $w$, the pair*

$$(\mathrm{Rec}(w, h), \mathrm{Rec}(-w, h^*))$$

*is $(0, 1)$ or $(1, 0)$ when the two distances at $(w, h)$ differ, and is $(0, 0)$ exactly when they tie. The tied residue is $w = (2h - 15)a \pmod q$.*

The proof of Proposition 4 uses the circular-center symmetry.

*Proof.* Substituting the center formulas gives both reflection identities. Circular distance is unchanged by negation, so reflection swaps the two distances. Strict inequality therefore reverses the chosen bit, whereas a tie is assigned to zero in both orientations. The arc of length $32a$ has one integer midpoint, at $c_0 - 16a = (2h - 15)a$. The other arc has odd length $32a + 1$ and has no integer midpoint. Thus this is the unique tied residue for that hint. □

This explains the mechanism more precisely than saying that two orientations uniformly double the quantizer resolution. Away from exact ties, the reflected observations are complements after relabeling the hint. Their extra discriminatory power arises at the integer boundary points. At gain $a$, the odd values of $v$ land exactly on such boundaries, while the adjacent even values do not. The deterministic tie rule is therefore part of the theorem, not an implementation detail to suppress.

**Supplementary evaluation records** The evaluation artifact also retains 80 compiled reference-C recoveries on 50 distinct additional keys using fixed-seed, OS-random, paired, and three-strategy configurations. These records verify the implementation in additional settings but do not contribute to the primary paired mean or confidence interval. Table 9 summarizes their call counts and honest-key predictions alongside the primary matched evaluation.

**Table 9:** Supplementary implementation configurations and the matched-key evaluation. The first eight rows report 80 supplementary recoveries on 50 distinct keys; the final three rows summarize the principal 150-recovery matched evaluation. Every execution recovers all 1,536 coefficients. *Pred.* counts successful honest-key predictions per strategy.

| Cohort | Keys | Calls/key | Mean | Pred. |
|---|---|---|---|---|
| Public seeds, orig. | 10 | 6,117–6,129 | 6,124.2 | 200/200 |
| OS-random, orig. | 10 | 6,120–6,135 | 6,127.5 | 200/200 |
| OS-random, opt. | 10 | 4,316–4,420 | 4,382.3 | 200/200 |
| Paired, orig. | 10 | 6,102–6,135 | 6,123.3 | 200/200 |
| Paired, opt. | 10 | 4,313–4,422 | 4,372.8 | 200/200 |
| Three-way orig. | 10 | 6,123–6,135 | 6,130.5 | 200/200 |
| Three-way opt. | 10 | 4,371–4,405 | 4,386.0 | 200/200 |
| Three-way graph | 10 | 3,540–3,586 | 3,565.7 | 200/200 |
| Matched orig. | 50 | 6,105–6,141 | 6,127.38 | 1,000/1,000 |
| Matched opt. | 50 | 4,330–4,421 | 4,382.54 | 1,000/1,000 |
| Matched graph | 50 | 3,525–3,602 | 3,565.18 | 1,000/1,000 |

**Table 10:** Supplementary reference-C implementation configurations (Windows 11, i7-10700F CPU, 48 GB RAM): per-key ordinary Decaps query counts across five ten-key groups. These results are excluded from the primary 50-key paired statistics.

| Key index | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Public seeds | 6,126 | 6,129 | 6,117 | 6,123 | 6,126 |
| OS-random orig. | 6,135 | 6,120 | 6,126 | 6,135 | 6,126 |
| OS-random opt. | 4,383 | 4,372 | 4,420 | 4,316 | 4,405 |
| Paired orig. | 6,129 | 6,111 | 6,135 | 6,126 | 6,102 |
| Paired opt. | 4,422 | 4,396 | 4,375 | 4,405 | 4,380 |
| Three-way orig. | 6,129 | 6,135 | 6,135 | 6,132 | 6,129 |
| Three-way opt. | 4,371 | 4,388 | 4,395 | 4,396 | 4,374 |
| Graph peel | 3,544 | 3,567 | 3,586 | 3,540 | 3,569 |
| Cycle edges | 22 | 36 | 36 | 6 | 46 |
| Key index | 5 | 6 | 7 | 8 | 9 |
| Public seeds | 6,120 | 6,129 | 6,123 | 6,129 | 6,120 |
| OS-random orig. | 6,123 | 6,126 | 6,129 | 6,132 | 6,123 |
| OS-random opt. | 4,379 | 4,386 | 4,357 | 4,395 | 4,410 |
| Paired orig. | 6,129 | 6,132 | 6,126 | 6,108 | 6,135 |
| Paired opt. | 4,365 | 4,313 | 4,369 | 4,375 | 4,328 |
| Three-way orig. | 6,132 | 6,135 | 6,132 | 6,123 | 6,123 |
| Three-way opt. | 4,374 | 4,390 | 4,376 | 4,391 | 4,405 |
| Graph peel | 3,573 | 3,568 | 3,560 | 3,586 | 3,564 |
| Cycle edges | 32 | 32 | 40 | 28 | 12 |

**Table 11:** Per-key results for the principal matched evaluation (Ubuntu, AMD Ryzen AI 9 HX 370, 32 GB RAM): 50 matched keys, source-identical three-way owner/client recoveries under GCC `-O3`. Each half lists 25 keys; *Orig.*, *Opt.*, and *Graph* are actual Decaps counts, and *Cyc.* counts distinct edges in the three unbalanced fundamental cycles. Full raw records are in the extension artifact.

| Key | Orig. | Opt. | Graph | Cyc. | Key | Orig. | Opt. | Graph | Cyc. |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 6,141 | 4,351 | 3,594 | 30 | 25 | 6,105 | 4,372 | 3,558 | 46 |
| 1 | 6,120 | 4,376 | 3,572 | 36 | 26 | 6,138 | 4,388 | 3,578 | 18 |
| 2 | 6,138 | 4,411 | 3,590 | 42 | 27 | 6,120 | 4,382 | 3,595 | 38 |
| 3 | 6,123 | 4,368 | 3,543 | 24 | 28 | 6,129 | 4,388 | 3,577 | 20 |
| 4 | 6,123 | 4,398 | 3,554 | 36 | 29 | 6,126 | 4,371 | 3,562 | 28 |
| 5 | 6,132 | 4,336 | 3,531 | 14 | 30 | 6,138 | 4,389 | 3,565 | 20 |
| 6 | 6,111 | 4,412 | 3,560 | 46 | 31 | 6,111 | 4,395 | 3,559 | 26 |
| 7 | 6,132 | 4,372 | 3,560 | 18 | 32 | 6,123 | 4,383 | 3,547 | 22 |
| 8 | 6,123 | 4,330 | 3,563 | 16 | 33 | 6,132 | 4,385 | 3,544 | 22 |
| 9 | 6,135 | 4,357 | 3,561 | 16 | 34 | 6,129 | 4,395 | 3,545 | 30 |
| 10 | 6,126 | 4,378 | 3,563 | 54 | 35 | 6,129 | 4,412 | 3,579 | 16 |
| 11 | 6,138 | 4,405 | 3,585 | 16 | 36 | 6,129 | 4,421 | 3,574 | 8 |
| 12 | 6,120 | 4,404 | 3,570 | 20 | 37 | 6,138 | 4,364 | 3,559 | 22 |
| 13 | 6,114 | 4,409 | 3,564 | 16 | 38 | 6,138 | 4,384 | 3,585 | 18 |
| 14 | 6,138 | 4,345 | 3,564 | 32 | 39 | 6,126 | 4,368 | 3,540 | 26 |
| 15 | 6,129 | 4,389 | 3,581 | 72 | 40 | 6,132 | 4,396 | 3,588 | 12 |
| 16 | 6,126 | 4,374 | 3,549 | 12 | 41 | 6,129 | 4,352 | 3,525 | 18 |
| 17 | 6,123 | 4,372 | 3,579 | 24 | 42 | 6,126 | 4,374 | 3,571 | 10 |
| 18 | 6,114 | 4,407 | 3,548 | 8 | 43 | 6,132 | 4,349 | 3,562 | 16 |
| 19 | 6,132 | 4,384 | 3,585 | 22 | 44 | 6,120 | 4,381 | 3,564 | 28 |
| 20 | 6,129 | 4,388 | 3,545 | 20 | 45 | 6,138 | 4,390 | 3,549 | 28 |
| 21 | 6,132 | 4,386 | 3,583 | 28 | 46 | 6,129 | 4,410 | 3,602 | 34 |
| 22 | 6,120 | 4,394 | 3,529 | 16 | 47 | 6,123 | 4,390 | 3,555 | 24 |
| 23 | 6,123 | 4,357 | 3,587 | 16 | 48 | 6,132 | 4,418 | 3,574 | 14 |
| 24 | 6,123 | 4,388 | 3,554 | 22 | 49 | 6,132 | 4,379 | 3,588 | 58 |