Signing-Key Recovery from Unsalted Root Expansion and Salt-Binding Repair for MQOM v2

José Luis Delgado

Independent Researcher

Abstract. We give the first passive signing-key recovery attack on MQOM v2 in the classical EUF-CMA setting by exploiting the generation of the correlated GGM trees used in each signature. MQOM v2 expands a fresh λ-bit signature master seed, denoted mseed in the specification, with a PRG call whose salt input is always zero. A repeated mseed therefore produces the same GGM root seeds $\mathsf{rseed}[e]$, even when the signatures have different salts. For each execution, the BLC opening contains a sibling path whose top seed is either $\mathsf{rseed}[e]$ or its XOR with the first λ bits of the secret MQ solution. When two signatures repeat mseed and the most significant bits of their hidden-leaf indexes differ, XORing the two top seeds cancels $\mathsf{rseed}[e]$ and reveals this secret prefix. The attacker can then recover a GGM root seed, reconstruct the corresponding tree and the secret MQ solution, and sign a fresh message.
The value mseed is hidden, but the BLC openings provide enough information to detect its repetition. Storing the top sibling-path seeds from the first two executions together with their XOR finds every collision that can reveal the secret prefix. At the Category-I limit of 2^{64} signing queries, the attack succeeds with probability about 0.39. The same zero-salt derivation also allows computations to be reused across public keys, parameter variants, and v2 releases, which yields the cross-key, exhaustive search, and precomputation extensions studied after the single-key attack. Executable tests against the official implementation recover signing keys and verify fresh-message forgeries. MQOM v3.0 binds root-seed derivation to the signature salt and execution index, removing the relation used by these attacks.

Artifact. The code and scripts for reproducing the attack, cost estimates, and salt-binding repair are available at https://github.com/hypergalois/MQOMRootExpansion.

1 Introduction

MQOM v2 is an MPC-in-the-Head signature scheme built from the Threshold-Computation-in-the-Head (TCitH) framework and correlated GGM trees [BBFR25, BFR24]. NIST advanced MQOM to the third round of the Additional Digital Signatures process in May 2026 [ABC+26], where signatures are evaluated against adaptive chosen-message attacks with up to 2^{64} signing queries in the security-strength estimates [Nat26, Nat22].

MQOM derives the array of GGM root seeds rseed for one signature from a single λ-bit signature master seed mseed. The v2 BLC commitment algorithm passes zero as the salt input when it expands mseed into this array, while the fresh signature salt is used only when the GGM trees are expanded from their root seeds. Consequently, repeating mseed repeats every $\mathsf{rseed}[e]$, even across signatures with different salts and public keys.

The BLC opening makes this reuse exploitable. At the top of each correlated GGM tree, one node is $\mathsf{rseed}[e]$ and the other is $\mathsf{rseed}[e] \oplus \mathsf{FirstBits}_\lambda(x)$, where x is the secret

E-mail: jose@delgado.fyi (José Luis Delgado)

MQ solution. The most significant bit of the hidden-leaf index determines which of these two node seeds is the last element of the sibling path, so each execution discloses either $\mathsf{rseed}[e]$ or $\mathsf{rseed}[e] \oplus \mathsf{FirstBits}_\lambda(x)$.

Consider two signatures that use the same mseed and therefore have the same GGM root seed in every execution. Whenever the corresponding most significant bits of the hidden-leaf indexes differ, XORing the last seeds in the two sibling paths cancels the common root seed and leaves $\mathsf{FirstBits}_\lambda(x)$. From this prefix, the attacker recovers a root seed, expands its GGM tree under the public signature salt, reconstructs the secret MQ solution from the published correction value, and signs a fresh message.

Since the signature master seed is not published, the attacker recognizes its repetition through the BLC openings. Let T_0 and T_1 be the top sibling-path seeds from the first two executions, and store T_0, T_1, and $T_0 \oplus T_1$. Two signatures with the same mseed match in at least one of these three values: in the first value when their first top-level bits agree, in the second when their second top-level bits agree, and in the XOR when both bits differ. Including the corresponding bit or parity in the index therefore finds every repeated seed with three stored values per signature. The full vectors of top-level bits determine whether the collision reveals the secret prefix, and the public MQ instance rejects accidental matches.

Category-I signature master seeds have 128 bits, so a repetition occurs with constant probability at the permitted boundary of 2^{64} signatures. For both the short and fast parameter sets, the resulting attack recovers a full key and forges with probability about 0.39, using classical signing queries, expected linear-time indexing, and three $O(\lambda)$-bit records per signature. The precise probabilities and their error term are derived in Section 3.2.

Root-seed derivation also omits the public key, message, field, and PIOP round count, allowing the same evaluations to be reused across targets and releases. After developing the collision attack under one public key, we apply the same recovery step to collisions between unrelated keys, exhaustive search in every security category, precomputation tables, and distinguished-point time–memory tradeoffs.

1.1 Results

The main result is the single-key attack described above. Sections 2 and 3 derive the opening equation, use it to recover the secret MQ solution, and compute the Category-I success probability; within the indexing model used by the attack, they also prove that three stored values per signature are necessary.

The remaining results are extensions of the same zero-salt root-seed derivation. Section 4 treats collisions between different public keys, while Section 5 develops exhaustive search and precomputation from the same recovery equation. Section 8 validates the attack against the official code, and Section 9 gives the salt- and execution-dependent root-seed derivation adopted in MQOM v3.0.

2 Top seed in a correlated GGM tree opening

The example above depends on one relation between a BLC opening and the secret MQ solution, which we now express in the notation of the specification. Fix a public key with secret MQ solution x. For a signature s, let $m_s \in \{0,1\}^\lambda$ denote the signature master seed mseed and let $\mathsf{salt}_s \in \{0,1\}^\lambda$ be the signature salt. The v2 BLC commitment algorithm derives the GGM root-seed array as

$$(r_{s,0}, \ldots, r_{s,\tau-1}) \leftarrow \mathsf{Parse}\big(\mathsf{PRG}(0^\lambda, 0, m_s, \tau\lambda)\big). \tag{1}$$

At execution e, set

$$r_{s,e} = H_e(m_s), \qquad \delta = \mathsf{FirstBits}_\lambda(x).$$

Thus $r_{s,e}$ denotes the specification value $\mathsf{rseed}[e]$ for signature s. The public function H_e returns component e of Equation (1), and δ is the first λ bits of the secret MQ solution. Algorithm 13 of the v2.1 specification initializes the two top nodes of the correlated GGM tree as

$$\mathsf{node}_{s,e}[2] = r_{s,e}, \qquad \mathsf{node}_{s,e}[3] = r_{s,e} \oplus \delta.$$

The signature salt enters only the lower tree levels, while both top nodes depend only on m_s and the fixed value δ.

Let $i^*_{s,e} \in [0, N-1]$ denote the hidden-leaf index for execution e. Algorithm 14 serializes the sibling path $\mathsf{path}[e]$ from the leaf toward the root. With $\ell = \log_2 N$, write $T_{s,e} = \mathsf{path}_s[e][\ell-1]$ for its last seed. The most significant bit of $i^*_{s,e}$ identifies the top-level subtree containing the hidden leaf, and therefore determines which of the two nodes above is disclosed as its sibling.

Lemma 1 (Top seed in a sibling path)**.** *For every valid v2.1 signature and execution,*

$$T_{s,e} = r_{s,e} \oplus c_{s,e}\delta, \qquad c_{s,e} = 1 - \mathsf{MSB}(i^*_{s,e}). \tag{2}$$

A hidden leaf in the right half therefore exposes $r_{s,e}$, whereas a hidden leaf in the left half exposes $r_{s,e} \oplus \delta$.

Proof. When $\mathsf{MSB}(i^*) = 0$, the hidden path begins below node 2, so its top-level sibling is node 3 and equals $r \oplus \delta$. When $\mathsf{MSB}(i^*) = 1$, node 2 is the sibling and equals r. Algorithm 14 places this sibling seed last in $\mathsf{path}[e]$, with the same orientation used by the official C implementation. □

Verification recomputes the hidden-leaf index array i^*, so the attacker knows every $c_{s,e}$ and reads every $T_{s,e}$ from the BLC opening. The only unknown values in Equation (2) are the GGM root seed $r_{s,e}$ and the secret prefix δ shared by all executions under the same key.

3 Signing-key recovery from a repeated signature master seed

Suppose that two signatures $s \neq t$ use the same signature master seed. The zero-salt expansion is deterministic, so their GGM root seeds agree in every execution even when the signatures have different salts:

$$r_{s,e} = r_{t,e} \quad \text{for every } e.$$

XORing the two top sibling-path seeds then cancels this common root seed and gives

$$T_{s,e} \oplus T_{t,e} = (c_{s,e} \oplus c_{t,e})\delta. \tag{3}$$

Equation (3) gives zero when the top-level bits agree and gives δ directly when they differ. A colliding pair is therefore useful unless its two coefficient vectors $(c_{s,e})_e$ and $(c_{t,e})_e$ are identical, an event of probability $2^{-\tau}$.

Once δ is known, either signature reveals all its GGM root seeds:

$$r_{s,e} = T_{s,e} \oplus c_{s,e}\delta. \tag{4}$$

The attacker expands a recovered root seed under the signature salt and regenerates the leaf seeds and their expanded values in that tree. Let $\bar{x}_{s,e,i}$ be the additive share obtained from leaf i. The correlated GGM tree satisfies

$$\bigoplus_{i=0}^{N-1} \mathsf{lseed}_{s,e,i} = \delta.$$

Every serialized share begins with its leaf seed, while $\delta = \mathsf{FirstBits}_\lambda(x)$. The correction value

$$\Delta_{x,s,e} = x \oplus \bigoplus_i \bar{x}_{s,e,i}$$

therefore has a zero λ-bit prefix. The signature omits this known prefix and publishes the remaining suffix $\Delta'_{x,s,e}$. The recovered tree and the public suffix determine the whole witness:

$$x = \left(0^\lambda \,\|\, \Delta'_{x,s,e}\right) \oplus \bigoplus_{i=0}^{N-1} \bar{x}_{s,e,i}. \tag{5}$$

Serialized addition is XOR because every MQOM base field has characteristic two, and the recovered solution is checked against the public MQ instance.

Theorem 1 (Key recovery and forgery)**.** *Two signatures with the same signature master seed and different coefficient vectors $(c_{s,e})_e$ determine the secret MQ solution x. From $\mathsf{sk} = (\mathsf{pk}, x)$, the attacker generates a valid signature on a fresh message. This is a full key-recovery attack in the classical EUF-CMA setting.*

Proof. At a coordinate with different top-level bits, Equation (3) gives δ, after which Equation (4) gives every GGM root seed of either signature. The signature salt and GGM-tree algorithms regenerate the leaf shares, and Equation (5) reconstructs x. The public MQ instance verifies the solution, and the specified signing algorithm uses (pk, x) and fresh randomness to sign a message that was not queried before. □

3.1 Detecting a repeated signature master seed

The value mseed is hidden, so equality cannot be tested directly. For each signature s, use the first two executions and store

$$(0, c_{s,0}, T_{s,0}), \qquad (1, c_{s,1}, T_{s,1}), \qquad (01, c_{s,0} \oplus c_{s,1}, T_{s,0} \oplus T_{s,1}). \tag{6}$$

In each record, the first component identifies its type, the second contains the corresponding top-level bit or parity, and the third is the indexed value.

For two signatures s and t with the same mseed, the first records match when $c_{s,0} = c_{t,0}$, and the second records match when $c_{s,1} = c_{t,1}$. In the remaining case both top-level bits are reversed, so $c_{s,0} \oplus c_{s,1} = c_{t,0} \oplus c_{t,1}$ and their third records match. Since these cases cover every pair of two-bit vectors, the three records detect every repeated seed, and a detected pair yields a key whenever its coefficient vectors $(c_{s,e})_e$ and $(c_{t,e})_e$ are not identical.

Candidate pairs can be checked before reconstruction by requiring one value $\widehat{\delta}$ to satisfy all τ equations

$$T_{s,e} \oplus T_{t,e} = (c_{s,e} \oplus c_{t,e})\widehat{\delta}.$$

This test removes accidental bucket matches when the sibling-path seeds are retained. The compact implementation below may discard them and check the recovered solution against the public MQ instance.

A colliding pair with identical coefficient vectors yields no equation for δ, whereas complementary coefficient vectors yield δ at every coordinate. Since the top-level bits are independent and uniform, the probability that a colliding pair is useful is

$$p_\tau = 1 - 2^{-\tau}. \tag{7}$$

Theorem 2 (Detection with three stored values)**.** *Given Q valid signatures under one public key, the construction in Equation (6) uses $3Q$ indexed records. It uses only two executions and detects every collision in* mseed*. Each pair with different coefficient vectors yields δ. Consider any method that stores a fixed XOR from a nonempty subset of the public root-seed coordinates and indexes it by that subset and by the parity of the corresponding top-level bits. Such a method needs at least three stored values for each coefficient vector, even if it chooses the subsets after seeing that vector.*

Proof. The argument above proves detection. To prove the lower bound, consider a nonempty XOR subset $A \subseteq [\tau]$. Its indexed record is labelled by the parity $\bigoplus_{e\in A} c_e$. For one fixed coefficient vector, that label agrees with exactly $2^{\tau-1}-1$ of the other $2^\tau-1$ vectors. Two records can therefore cover at most $2^\tau - 2$ other vectors, leaving at least one pair undetected. Every vector therefore needs at least three records, and Equation (6) meets this bound. For each detected pair with different coefficient vectors, Equation (3) returns δ on every differing coordinate and satisfies the check against all executions. Hashing gives constant expected bucket work per record, while external sorting gives a low-RAM implementation. □

Under the random-output model, the expected number of accidental bucket matches between unrelated signatures before the vector check is obtained by considering the three record types separately. For each type, two independent top-level-bit labels agree with probability $1/2$, after which their λ-bit indexed values agree with probability $2^{-\lambda}$. Hence

$$\mathbb{E}[C_{\text{false}}] = \frac{3Q(Q-1)}{4\,2^\lambda}. \tag{8}$$

At $Q = 2^{64}$ this is $0.75 + O(2^{-64})$, independent of short/fast choice.

The number of stored values can be reduced when lower detection probability is acceptable. One value detects half of all seed collisions, two values detect three quarters, and the three values in Equation (6) detect all of them. In every case, identical coefficient vectors remain unusable with probability $2^{-\tau}$. A sparser variant stores $T_{s,0}$ only when $c_{s,0} = 0$. It uses $Q/2$ records on average and has useful-pair probability $\frac{1}{4}(1 - 2^{1-\tau})$. Table 1 gives the resulting Category-I short tradeoff.

Table 1: Storage and success probability for the one-key attack at $Q = 2^{64}$ and $\tau = 12$.

Detector	records	usable probability per repeated seed	success
sparse first value	$Q/2$	0.249878	0.117449
one value	Q	0.499756	0.221104
two values	$2Q$	0.749756	0.312627
three values	$3Q$	0.999756	0.393395

The three-value detector gives the largest success at fixed Q, while the sparse variant reduces the average index to half a record per signature and retains success above 0.117. Both use chosen-message signatures under one public key and stay within the 2^{64} query boundary.

3.2 Category-I success probability

For independent uniform λ-bit signature master seeds, the mean number of useful colliding pairs among Q signatures is

$$\mu(Q, \lambda, \tau) = \binom{Q}{2} 2^{-\lambda}\big(1 - 2^{-\tau}\big). \tag{9}$$

In the birthday regime $Q = O(2^{\lambda/2})$, triple occupancies contribute only $O(Q^3/2^{2\lambda})$. Poissonization therefore gives

$$\Pr[\text{key recovery}] = 1 - e^{-\mu} + O(Q^3/2^{2\lambda}). \tag{10}$$

The error term is asymptotic in λ and becomes $O(2^{-\lambda/2})$ at $Q = 2^{\lambda/2}$.

For Category I, $\lambda = 128$, and at $Q = 2^{64}$ the approximation error is $O(2^{-64})$ and $\binom{Q}{2}/2^{128} = 1/2 - 2^{-65}$. For the short parameters, where $\tau = 12$, the pair $(\mu, \Pr[\text{recovery}])$ is approximately $(0.499878, 0.393395)$. For the fast parameters, where $\tau = 17$, it is approximately $(0.499996, 0.393467)$. The argument applies to the $\mathbb{F}_2$, $\mathbb{F}_{16}$, and $\mathbb{F}_{256}$ families and to the three- and five-round PIOP variants because all of them use the same zero-salt root-seed derivation and the same top sibling-path relation.

NIST includes up to 2^{64} chosen-message signatures in its EUF-CMA security estimates and focuses primarily on classical signing-oracle queries [Nat26, Nat22]. The attack uses classical queries at this boundary and succeeds with constant probability. Its online data and indexing work are about 2^{64}, below the Category-I reference strength. The detector emits $3Q$ records of $O(\lambda)$ bits. Hashing processes them in expected linear time, and external sorting provides the low-RAM path. Table 2 gives packed storage costs.

3.3 Reducing collision-index storage

If the BLC openings are retained, a matching pair needs no additional algebra: any execution with different top-level bits gives δ directly through Equation (3). The index may instead discard the openings after processing each signature. In that case, one checksum per signature recovers a chosen prefix of δ. This reduces persistent storage at the cost of enumerating the remaining witness bits after a match.

Let π_b select the leading b bits, where $\tau < b \le \lambda$, and identify these bits with the polynomial basis of

$$\mathbb{F}_{2^b} = \mathbb{F}_2[X]/f_b(X)$$

for an irreducible polynomial f_b of degree b. Store with signature s the checksum

$$S_s^{(b)} = \sum_{e=0}^{\tau-1} X^e \pi_b(T_{s,e}). \tag{11}$$

This sum is evaluated in $\mathbb{F}_{2^b}$, so $S_s^{(b)}$ is one b-bit field element, not a list of τ coefficients. For two signatures with the same signature master seed, Equation (2) gives

$$S_s^{(b)} + S_t^{(b)} = \beta_{s,t}\pi_b(\delta), \qquad \beta_{s,t} = \sum_{e=0}^{\tau-1} (c_{s,e} \oplus c_{t,e}) X^e. \tag{12}$$

When the coefficient vectors differ, $\beta_{s,t}$ is a nonzero polynomial of degree less than $\tau < b$, so it is invertible. Therefore

$$\pi_b(\delta) = \beta_{s,t}^{-1}(S_s^{(b)} + S_t^{(b)}). \tag{13}$$

At $b = 128$, the checksum recovers all of δ. Storing $T_{s,0}$ as well then recovers $r_{s,0} = T_{s,0} \oplus c_{s,0}\delta$. The signature salt expands that entire tree, and the published correction-value suffix for execution zero then recovers the witness through Equation (5). One execution is therefore enough after the index reports a matching pair.

For Category-I GF(2) short parameters, direct reconstruction retains $S_s^{(128)}$ (16 bytes), $T_{s,0}$ (16), the salt (16), the 12-bit coefficient vector (2), and the correction suffix (4), for a total of 54 bytes. The GF(2) witness has 160 serialized bits. A b-bit checksum and the two-byte coefficient vector therefore suffice if the attacker enumerates the remaining $160 - b$ bits against the public MQ instance. With $b = 96$, this form uses 14 bytes and at most 2^{64} public MQ evaluations, matching the signing-query exponent.

An external-sort record contains a 16-byte indexed value. The recovery data can be stored inside each record or once in a separate array addressed by an eight-byte reference. The storage totals use the smaller layout, while physical partitions encode the record type and its top-level bit or parity. The resulting persistent sizes are:

Table 2: Packed Category-I GF(2) short storage at $Q = 2^{64}$. "Direct" uses a 54-byte recovery record with no search; "projected" uses $b = 96$ and a 2^{64} public completion.

Detector	success	direct B/input	direct ZiB	projected B/input	projected ZiB
sparse first value	0.117449	35	0.546875	15	0.234375
one value	0.221104	70	1.093750	30	0.468750
two values	0.312627	102	1.593750	60	0.937500
three values	0.393395	126	1.968750	86	1.343750

The sparse row is an average because only signatures with zero parity emit a record and retain recovery data. Intermediate projections give a continuous tradeoff: $b = 40, 64, 96, 128$ leaves respectively $120, 96, 64, 32$ unknown GF(2) witness bits. The sparse endpoint then occupies $0.179688, 0.203125, 0.234375$, and 0.265625 ZiB; the three-value detector occupies $1.078125, 1.21875, 1.34375$, and 1.40625 ZiB. Even the minimum stored $b = 40$ point requires $2^{120} < 2^{128}$ public evaluations and leaves the collision probability unchanged.

Table 2 counts packed persistent data. Temporary merge-sort copies, signing-oracle bandwidth, and signature generation are additional resources. Truncating an indexed value to b bits creates $3Q(Q-1)/(4\,2^b)$ expected candidates before verification; the table retains all 128 bits.

Theorem 3 (Recovery from a projected checksum)**.** *For Category-I GF(2), any $\tau < b \leq 128$ checksum in Equation (11) recovers b bits of the 160-bit witness and completes a signing key in at most 2^{160-b} public MQ evaluations. Detector coverage and the success probabilities in Table 1 remain unchanged.*

Proof. Equation (13) recovers the selected b bits and leaves a suffix of $160 - b$ bits. Exhaustive search against the public MQ instance includes the signer's witness and may stop at any satisfying assignment because each such assignment is a valid signing witness. This enumeration starts only after the three-value or sparse detector reports a candidate pair. □

4 Collisions between different public keys

The attack above uses two signatures under one public key, so both openings contain the same witness prefix δ. The zero-salt root-seed derivation also omits the public key, which allows the same signature master seed to collide across unrelated keys. In that case the two signatures contain different witness prefixes, but the common GGM root seeds still give a small linear system.

Let signatures A and B use distinct keys with witness prefixes δ_A and δ_B, and suppose they repeat a signature master seed. Their root-seed arrays agree because root-seed derivation is independent of the public key. Writing $a_e = c_{A,e}$ and $b_e = c_{B,e}$, the XOR of their top sibling-path seeds is

$$D_e = T_{A,e} \oplus T_{B,e} = a_e\delta_A \oplus b_e\delta_B. \tag{14}$$

The unknowns in Equation (14) are the two λ-bit prefixes. Two linearly independent coefficient rows recover both. The remaining rows and both public MQ instances then verify the result, after which Equation (5) reconstructs both signing keys.

4.1 Detecting cross-key collisions

A row of type $(a_e, b_e) = (0, 0)$ is visible because the two sibling paths contain the same root seed. To detect the other row types without knowing either witness prefix, we cancel pairs of coordinates inside each signature. Whenever $c_{s,e} = c_{s,f}$, define

$$V_{s,e,f} = T_{s,e} \oplus T_{s,f}.$$

If coordinates e and f have the same coefficient pair $(a_e, b_e) = (a_f, b_f)$, both witness prefixes cancel and

$$V_{A,e,f} = r_e \oplus r_f = V_{B,e,f}.$$

Five coordinates suffice when they are viewed as the vertices of K_5 and vertex e is colored by $(a_e, b_e) \in \mathbb{F}_2^2$. Any coloring with these four possible values has a monochromatic edge. Both signatures emit the same $V_{s,e,f}$ on that edge, so the index finds every pair that shares the same root-seed array.

Emitting every eligible edge of K_5 uses five records per signature on average. The public coefficient vector determines which edges a signature may emit, so a fixed schedule can reduce this number. Identify a coefficient vector with its complement, set its first bit to zero, and encode the remaining four bits little-endian as $c \in \{0, \ldots, 15\}$. Writing ij for edge (i, j) gives the schedule below.

Table 3: The 16-class K_5 schedule.

c	selected edges	c	selected edges
0	02, 04, 24	8	01, 02, 12
1	02, 04, 24	9	02, 03, 14, 23
2	01, 04, 14	10	01, 03, 13, 24
3	03, 04, 12, 34	11	12, 14, 24
4	01, 02, 12	12	01, 02, 12
5	02, 04, 24	13	02, 13, 14, 34
6	01, 04, 14	14	01, 23, 24, 34
7	04, 12, 13, 23	15	12, 14, 24

Every listed edge is monochromatic for its class, and every pair of rows in Table 3 shares an edge. These two properties, verified over all 32^2 ordered coefficient vectors, guarantee detection. The 54 listed edges use $27/8 = 3.375$ records per signature on average and at most four. The proposition below gives a matching lower bound. A candidate survives only if Equation (14) holds at every coordinate and both recovered witnesses satisfy their public MQ instances.

Proposition 1 (Minimum size of the 16-class schedule)**.** *Suppose every class c is assigned a set S_c of c-monochromatic K_5 edges and completeness requires $S_c \cap S_d \neq \varnothing$ for all c, d. Then $\sum_c |S_c| \geq 54$. Hence Table 3 meets this lower bound.*

Proof. For a class c, regard its selected edges S_c as a graph on five vertices. If S_c were bipartite, a proper binary coloring d would make every edge of S_c bichromatic. Since S_d may contain only edges monochromatic under d, the two schedules could not intersect. Thus every S_c is non-bipartite and has at least three edges.

Among the ten $2 + 3$ complement classes, the only allowed three-edge non-bipartite graph in each class is the triangle on its three-vertex color class. Two such forced triangles are edge-disjoint exactly when their three-vertex sets intersect in one vertex. Under complementation, the incompatibility graph is the Kneser graph $\mathsf{KG}(5, 2)$, i.e., the Petersen graph. An independent set in $\mathsf{KG}(5, 2)$ is a pairwise-intersecting family of edges of K_5; it is either a star of size at most four or a triangle of size three. Its independence number is therefore four, so every vertex cover has size at least $10 - 4 = 6$. At least six of the ten forced rows consequently need a fourth edge. Summing gives $16 \cdot 3 + 6 = 54$, attained by the table. □

Proposition 2 (Minimum fixed-edge detector)**.** *The ten edges of K_5 detect every four-type row coloring. No graph with fewer than ten edges can do so.*

Proof. Any graph forcing a monochromatic edge under four colors has chromatic number at least five and contains a 5-critical subgraph whose minimum degree is at least four. Such a subgraph has at least five vertices and, by the handshake lemma, at least ten edges, with K_5 attaining both bounds. □

To account for accidental index matches under the random-output model, let

$$P_{\times} = \binom{Q}{2} - \sum_{u=1}^{U} \binom{q_u}{2}$$

be the number of signature pairs drawn from different public keys. The expected number of accidental indexed-coordinate matches is

$$\mathbb{E}[C_{\times,\mathrm{false}}] = \frac{83 P_{\times}}{64\, 2^{\lambda}}. \tag{15}$$

The coefficient $83/64$ is the mean intersection size of two scheduled edge sets under independent public coefficient vectors. Indexing all pairs would give $\binom{\tau}{2}/4$, which equals 16.5 at $\tau = 12$, compared with 1.296875 for the schedule. The full linear system and the public MQ instances reject accidental matches.

Theorem 4 (Recovery of two unrelated keys)**.** *For $\tau \geq 5$, the scheduled K_5 index detects every cross-key collision in* mseed*. Conditional on a collision and independent uniform coefficient vectors, both signing keys are recovered with probability*

$$p_2(\tau) = 1 - 3 \cdot 2^{-\tau} + 2 \cdot 4^{-\tau}. \tag{16}$$

At least one signing key is recovered with probability

$$p_{\geq 1}(\tau) = 1 - 2^{-\tau}. \tag{17}$$

The expected number of recovered keys per colliding pair is $2p_2(\tau) + 2(2^{\tau} - 1)/4^{\tau}$.

Proof. A $\tau \times 2$ binary matrix has rank below two precisely when all rows lie in one of the three one-dimensional subspaces. Inclusion–exclusion counts $3 \cdot 2^{\tau} - 2$ such matrices, proving

Equation (16). In rank two, Equation (14) uniquely gives both deltas and Theorem 1 applies independently to the two signatures.

Of the rank-one matrices, the $2^\tau - 1$ nonzero sequences over $\{00, 10\}$ reveal δ_A, and the symmetric sequences over $\{00, 01\}$ reveal δ_B. These two families are disjoint. The third subspace $\{00, 11\}$ reveals only $\delta_A \oplus \delta_B$. Adding the two useful rank-one families to the rank-two count gives $p_{\geq 1} = 1 - 2^{-\tau}$, while detection follows from the repeated-row argument above and recovery of each available full key follows from Equations (4) and (5). □

Across U keys, the mean number of collision events that recover two keys is

$$\mu_\times = \frac{P_\times}{2^\lambda}\, p_2(\tau). \tag{18}$$

If Q signatures are divided equally among U keys, the leading fraction of cross-key pairs is $1 - 1/U$. At $\lambda = 128, Q = 2^{64}$, this fraction tends to one as U grows, and $\mu_\times \to p_2(\tau)/2$. Poissonization gives two-key recovery probabilities $1 - e^{-\mu_\times}$: approximately 0.393247 for $\tau = 12$ and 0.393462 for $\tau = 17$. Counting useful rank-one events raises the probability of at least one recovered key to the same limits as the one-key collision attack, approximately 0.393395 and 0.393467. The expected numbers of recovered keys tend to 0.999512 and 0.999985: nearly one compromised key in expectation from an aggregate 2^{64}-signature corpus.

The 54-edge schedule minimizes storage within the fixed five-coordinate complement-class model and detects every collision at fixed Q. If more public signatures are available, a sparse index can obtain more collision opportunities from the same storage. Let $S(c)$ be a deterministic set of pair-cancellation edges emitted for one public coefficient vector c, let $r = \mathbb{E}[|S(c)|]$, and let $d = \Pr[S(c_A) \cap S(c_B) \neq \varnothing]$ for independent coefficient vectors. If $p_e = \Pr[e \in S(c)]$, then

$$d \leq \sum_e p_e^2 \leq \left(\sum_e p_e\right)^2 = r^2. \tag{19}$$

The first inequality is a union bound over common edges, and the second uses non-negativity. With Q signatures and mean index volume $R = Qr$, the intensity of genuine shared-seed candidates is therefore at most $R^2/(2^{\lambda+1})$ in the birthday limit.

A single fixed edge (e, f) meets this bound by emitting $T_e \oplus T_f$ exactly when $c_e = c_f$. Here $r = 1/2$ and $d = 1/4 = r^2$. Conditioned on a genuine bucket collision, the two coefficient rows on that edge are equal and uniform, while the remaining $\tau - 2$ rows stay uniform. Direct counting gives

$$p_{2|\text{edge}}(\tau) = 1 - 3 \cdot 2^{1-\tau} + 2^{3-2\tau}, \tag{20}$$

$$p_{\geq 1|\text{edge}}(\tau) = 1 - 2^{1-\tau}. \tag{21}$$

Theorem 5 (Sparse cross-key detector)**.** *Among deterministic pair-cancellation schedules, the fixed-edge construction maximizes genuine collision detection per squared mean index record. Its two-key event mean is*

$$\mu_{\text{sparse}} = \frac{P_\times}{2^\lambda} \cdot \frac{1}{4}\left(1 - 3 \cdot 2^{1-\tau} + 2^{3-2\tau}\right). \tag{22}$$

At fixed Q, it detects one quarter of collisions, whereas the scheduled K_5 index detects every collision.

Proof. Equation (19) proves the upper bound. The fixed edge attains it because $d = 1/4$ and $r = 1/2$, hence $d/r^2 = 1$. For the recovery probability, put $n = \tau - 2$. If the repeated row is 00, the rank-deficient tails are the union of the three binary lines and number

$3 \cdot 2^n - 2$. For each of the three nonzero repeated rows, exactly 2^n tails remain in its line. Dividing $6 \cdot 2^n - 2$ by the 4^{n+1} conditioned row systems proves Equation (20). Failure to identify either individual key occurs only in the 00/11 line; its conditioned probability is $2^{-(n+1)} = 2^{1-\tau}$, proving Equation (21). Multiplying the shared-input pair count, detection probability, and rank-two probability gives Equation (22). □

For Category-I short parameters and $Q = 2^{64}$, the sparse index uses $Q/2$ records on average and has two-key success about 0.1173, compared with about 0.3932 for the K_5 schedule. At the latter's mean index volume $(27/8)2^{64}$, the sparse index can process $(27/4)2^{64}$ signatures, and its two-key event mean becomes 5.687, giving success about 0.9966. The event-rate gain over the K_5 schedule at equal index volume is about 11.38. This equal-volume comparison uses aggregate data across public keys; the single-key signing-query limit is 2^{64}.

Signatures from unrelated users enter the same birthday pool because root derivation omits both the public key and salt. Parameter variants with the same λ also share their common root prefix; the smaller value of τ determines the usable number of executions. NIST lists multi-key resistance among the desired properties [Nat22].

5 Recovery through the shared root functions

The collision attack uses two evaluations generated by the same hidden signature master seed. The zero-salt call also permits inversion of one shared root-seed function against root seeds collected from many signatures, which extends the attack to Categories III and V even when the signing-query limit is below their birthday bound.

For a fixed execution e_0, a signature with $c_{s,e_0} = 0$ discloses the GGM root seed

$$T_{s,e_0} = H_{e_0}(m_s).$$

The same signature must have $c_{s,e} = 1$ in some execution so that a recovered seed yields δ. Conditional on $c_{s,e_0} = 0$, recovery fails only when $c_{s,e} = 0$ for every other execution as well. Under uniform Fiat–Shamir challenges, one signature therefore supplies a usable target with probability

$$p_{\mathrm{tar}}(\tau) = \frac{1}{2} - 2^{-\tau}. \tag{23}$$

Thus q signatures provide $M_{\mathrm{tar}} \approx q p_{\mathrm{tar}}(\tau) \approx q/2$ targets. To test a candidate m, the attacker compares its root seeds with every coordinate for which $c_{s,e} = 0$. Every coordinate with $c_{s,e} = 1$ must produce the same δ. Equation (5) then reconstructs x, and the public MQ instance verifies it. A second disclosed root seed gives an inexpensive early filter.

5.1 Exhaustive search in every security category

Enumerate the 2^λ signature master seeds in a fixed order and look up $H_{e_0}(m)$ among the public targets. Every target is the image of an honest uniform seed. With M_{tar} target inputs, the mean number of candidates examined through the first honest input is at most

$$1 + \frac{2^\lambda}{M_{\mathrm{tar}} + 1}. \tag{24}$$

For M_{tar} independent uniform positions in a domain of size N, the tail sum satisfies

$$\sum_{j=1}^{N} (j/N)^{M_{\mathrm{tar}}} \le 1 + N/(M_{\mathrm{tar}} + 1).$$

This bound finds an honest target input for any deterministic H_{e_0}. Transcript checks reject other preimages of the same output. The random-output model predicts a constant number of additional validations at this scale; for a concrete H_{e_0}, the number follows its measured collision profile.

Theorem 6 (All-category classical key recovery)**.** *Given q signatures and M_{tar} usable targets, exhaustive search reaches an honest target seed within the expectation in Equation (24). Under the random-output estimate for false matches, it recovers a signing key in $O(2^\lambda/M_{\mathrm{tar}})$ evaluations of H_{e_0}, plus transcript processing, target storage, and polynomial factors. With $M_{\mathrm{tar}} \approx q(1/2 - 2^{-\tau})$, the leading search cost is*

$$T_{\mathrm{enum}} \approx 2^{\lambda+1}/q. \tag{25}$$

The attack applies to every MQOM v2 field and round variant.

Proof. The targets are images of independent uniform signature master seeds. Their positions in the enumeration order are independent uniform variables, so the preceding tail-sum argument bounds the first honest input by $1 + 2^\lambda/(M_{\mathrm{tar}} + 1)$. At that input, the sibling-path equations recover δ; Equation (5) reconstructs the witness, and the recovered solution satisfies the public MQ instance. The random-output estimate gives constant mean validation work at the search scale. Substituting $M_{\mathrm{tar}} = q(1/2 - 2^{-\tau})$ yields Equation (25). The argument uses only the sibling-path relation and witness reconstruction shared by every listed variant. □

At $q = 2^{64}$, target storage has leading exponent 63. For the short parameters, the search exponents in Categories I, III, and V are 65.0007, 129.0000, and 193.0000, respectively. The Category-III and Category-V attacks reduce the classical exponent by about 63 bits relative to exhaustive search over the λ-bit seed domain. In Category I, the birthday collision attack is stronger and reduces total work to about 2^{64}.

If a table indexes k root seeds, each seed candidate costs k evaluations. The probability that a signature has an indexed coordinate with $c_{s,e} = 0$ and some coordinate with $c_{s,e} = 1$ is $1 - 2^{-k} - 2^{-\tau}$. The resulting factor $k/(1 - 2^{-k} - 2^{-\tau})$ is minimized only at $k = 1$, as shown in Theorem 8.

5.2 Reusable precomputation tables

Each H_e uses zero salt and fixed public selectors. It does not depend on the public key, message, signature salt, field, or PIOP round count. At fixed λ, one precomputation therefore applies to different public keys and to compatible short and fast variants.

Define three functions from the first three executions,

$$G_{01}(m) = H_0(m) \oplus H_1(m), \quad G_{12}(m) = H_1(m) \oplus H_2(m), \quad G_{02}(m) = H_0(m) \oplus H_2(m).$$

Choose a set $A \subseteq \{0,1\}^\lambda$ of P signature master seeds and store $(i, j, G_{ij}(m)) \mapsto m$ for all three triangle edges and every $m \in A$. For a signature, color vertex $e \in \{0,1,2\}$ by c_e. Some triangle edge (i, j) has equal endpoint colors, and on that edge

$$T_i \oplus T_j = H_i(m) \oplus H_j(m) = G_{ij}(m). \tag{26}$$

The first such edge in lexicographic order determines one table lookup. If it identifies m, any coordinate with $c_f = 1$ gives $\delta = T_f \oplus H_f(m)$, after which Theorem 1 yields the signing key. The all-one vector is usable; only the all-zero vector leaves no equation for δ.

Theorem 7 (Three-edge precomputation table)**.** *Under the Fiat–Shamir challenge distribution, the event in which the honest signature master seed belongs to a three-edge table of P inputs and the signature yields its signing key has probability*

$$h_\triangle(P, \tau) = \frac{P}{2^\lambda}\left(1 - 2^{-\tau}\right). \tag{27}$$

For Q independent signatures the probability of at least one such compromise is

$$1 - \left(1 - h_{\triangle}(P, \tau)\right)^{Q}. \tag{28}$$

If public key u contributes q_u signatures, the expected number of distinct keys compromised through this event is

$$\sum_{u} \left[1 - \left(1 - h_{\triangle}(P, \tau)\right)^{q_u}\right]. \tag{29}$$

These expressions lower-bound total key-recovery probability and expected yield because collisions between stored root values can produce additional candidates. Preprocessing evaluates H_0, H_1, and H_2 and stores three records per input. Online processing makes one lookup per signature; only reported candidates require checks against all executions and the public MQ instance.

Proof. The honest signature master seed lies in A with probability $P/2^{\lambda}$, and Equation (26) gives a lookup for every binary coloring of the triangle. Conditional on a table hit, key recovery fails exactly when all τ coefficients are zero, with probability $2^{-\tau}$. This proves Equation (27). Independent trials give Equation (28); applying the same expression separately to each public key and linearity of expectation give Equation (29). The construction stores precisely the three triangle edges for every input. □

The event $m \in A$ does not require a distributional assumption about the root functions. Collisions among the indexed functions produce multiple candidates, which the remaining root-seed equations and the public MQ instance filter. Under a random-output estimate, the mean number of examined candidates is $QP/2^{\lambda}$ because each signature makes one lookup in an edge-specific table.

Within the pairwise-cancellation family, the fixed edge graph must contain a monochromatic edge under every binary coloring. This is equivalent to non-bipartiteness, and the smallest such graph is a triangle. A table that stores all τ root seeds directly uses τP records and has useful-vector probability $1 - 2^{1-\tau}$. For Category-I short parameters at $P = Q = 2^{64}$, that table uses $12P$ records and succeeds with probability about 0.631941. The triangle uses $3P$ records and succeeds with probability about 0.632031, with Poisson-limit error $O(2^{-64})$. The splits $(\log P, \log Q) = (80, 48)$ and $(96, 32)$ preserve the leading success while trading offline precomputation for online signatures.

Now fix a preprocessing budget of B stored and evaluated root seeds. A table that indexes k coordinates for each of $P = B/k$ inputs has useful-vector probability $1-2^{-k}-2^{-\tau}$ per covered input. Its useful probability per record is therefore

$$\eta_k(\tau) = \frac{1 - 2^{-k} - 2^{-\tau}}{k}. \tag{30}$$

Theorem 8 (Single-coordinate precomputation)**.** *For the event that the signature master seed is stored in the table, $\tau \geq 3$, $1 \leq k \leq \tau$, and a record budget $B \leq 2^{\lambda}$, $\eta_k(\tau)$ is maximized only by $k = 1$. This allocation stores one fixed root seed, say $H_0(m)$, for B distinct signature master seeds. For one signature, the probability that the seed is stored and the opening reveals the key is*

$$h_{\mathrm{wide}}(B, \tau) = \frac{B}{2^{\lambda}} \left(\frac{1}{2} - 2^{-\tau}\right). \tag{31}$$

It uses B records, B evaluations of H_0, and makes a lookup on half of signatures on average.

Proof. The indexed coordinate must have coefficient zero, while at least one of the τ coordinates must have coefficient one; these disjoint failure events give $1 - 2^{-k} - 2^{-\tau}$. For $k \le \tau$,

$$k\eta_1 - k\eta_k = \frac{k}{2} - 1 + 2^{-k} - (k-1)2^{-\tau} \ge 0,$$

with equality only for $k = 1$: for $k = 2$ the difference is $1/4 - 2^{-\tau} > 0$, while for $k \ge 3$ and $\tau \ge k$ it is at least $k/2 - 1 - (k-2)2^{-k} > 0$. For $B \le 2^\lambda$, each record should therefore be assigned to a distinct input. Membership and coefficient-vector independence give Equation (31). □

The same result holds when a record stores any fixed nonempty XOR of root-seed coordinates, including a single root seed. Its public counterpart cancels δ on exactly one parity half of the coefficient vectors; after excluding the all-zero vector, one record is useful on fraction $1/2 - 2^{-\tau}$. For any collection of records assigned to the same input, the union bound therefore caps useful probability per record by this same value. Storing one root seed attains the bound with one evaluation of a root-seed function. An XOR of two root seeds ties in stored-record efficiency but requires two evaluations. Thus the single-coordinate table maximizes successful table lookups under the joint record and root-evaluation budget within the fixed XOR-cancellation family. Validated collisions in the concrete root functions may provide additional candidates.

To compare both tables at the same resources, take $P = 2^{64}$ for the triangle and set $B = 3P$ for the single-coordinate table. At $Q = 2^{64}$, the corresponding event has mean 6141/4096 and Bernoulli success about 0.776706. The triangle has mean 4095/4096 and success about 0.632031. The triangle recognizes every stored input for every coefficient vector, whereas the single-coordinate allocation recovers more keys per unit of preprocessing.

5.3 Reducing memory with distinguished-point chains

The single-coordinate table stores one record for every precomputed signature master seed. Iterated chains reduce this memory by representing many consecutive seeds with one endpoint [Hel80]. Fix H_0 and form M chains of length L:

$$x_{a,0} \xrightarrow{H_0} x_{a,1} \xrightarrow{H_0} \cdots \xrightarrow{H_0} x_{a,L}, \qquad 0 \le a < M.$$

The public chain identifier a regenerates the start $x_{a,0}$, so the table stores only endpoint information and a. Given a published target $y = H_0(m)$, the attacker iterates H_0 from y for at most L steps. If the walk reaches a stored endpoint, regenerating the chain from its start recovers the preceding input m. The remaining root seeds and the public MQ instance check the candidate. A false endpoint fingerprint adds work but cannot produce an incorrect key.

Let $\mathcal{C}$ contain the distinct pre-endpoint states $x_{a,j}$ for $0 \le j < L$, and write $C = |\mathcal{C}|$. This measured quantity is the number of signature master seeds represented by the table for the concrete function H_0.

Theorem 9 (Recovery conditional on chain coverage)**.** *For Q independent signatures with uniform signature master seeds and independent uniform Fiat–Shamir challenges, the probability that a chain table with actual coverage C contains at least one usable honest signature master seed is exactly*

$$1 - \left(1 - \left(\frac{1}{2} - 2^{-\tau}\right)\frac{C}{2^\lambda}\right)^Q. \tag{32}$$

Every such event recovers a signing key. Equation (32) therefore lower-bounds total recovery probability, conditional on C, for any deterministic H_0 in the stated Fiat–Shamir model.

Collisions among full root-seed arrays can only add successes. Proposition 3 shows how stopping each chain at its first distinguished point measures C. Preprocessing work spent on starts that are capped or merged is counted separately. Processing the targets uses at most $QL(1/2 - 2^{-\tau})$ root-seed evaluations in expectation and requires only the endpoint table plus one live signature record. This bound allows any deterministic H_0 and arbitrary chain merging.

Proof. A signature succeeds exactly when its uniform signature master seed lies in $\mathcal{C}$, $c_{s,0} = 0$, and $c_{s,e} = 1$ for at least one other execution. Under the theorem's Fiat–Shamir model these independent events have probability $(C/2^{\lambda})(1/2 - 2^{-\tau})$. Independent signatures give Equation (32). Starting from $y = H_0(x_{a,j}) = x_{a,j+1}$ reaches the stored endpoint; reconstruction then recovers $x_{a,j}$. The preprocessing and online walk bounds follow directly. □

The success calculation uses the measured value C. A separate random-map estimate describes how strongly sparse chains are expected to merge:

$$\rho_{\text{merge}} = \frac{ML^2}{2^{\lambda}}. \tag{33}$$

Values far below one predict that shared suffixes are a lower-order effect. Preprocessing measures C for the constructed table; Equation (33) is used only to estimate merge load.

With a b-bit endpoint fingerprint, random false matches add reconstruction work in the ratio

$$\rho_{\text{fp}} = \frac{ML}{2^{b}} \tag{34}$$

relative to target walks. Every candidate is still validated. A bit-packed record needs $b + \lceil \log_2 M \rceil$ bits because the public chain identifier regenerates the λ-bit start.

A fixed-length chain requires an endpoint lookup after every online state. Distinguished points reduce this to one large-table access per target walk [Din14, LH12]. We stop at the first state that satisfies a public predicate D. Starts that fail to reach such a state before the cap are discarded, and one representative is kept for each distinct endpoint.

Proposition 3 (Coverage from first distinguished points)**.** *Let a retained chain contain all states before its first distinguished endpoint. Retained representatives with distinct endpoints have pairwise disjoint covered-state sets. Consequently, the sum of their lengths equals the number of distinct covered states.*

Proof. A chain terminating at its first distinguished point cannot repeat a nondistinguished state because, after a repeat, it would remain in a cycle without a later distinguished state. If two terminating chains share a state, determinism makes their suffixes identical and sends them to the same first distinguished endpoint, so distinct endpoints imply disjoint chains. □

Measured coverage determines success, while the large-instance cost model uses a random-map estimate for walk lengths. Online, each usable target is walked to its first distinguished point or discarded at the public cap. A terminating walk makes one query to a minimal perfect hash function (MPHF) and an independent tag. Its temporary record contains the table slot, walk distance, sibling-path seed, execution index, and a target fingerprint. When the bounded buffer fills, the precomputed chains are regenerated and matching endpoint groups are replayed. The public chain identifier regenerates each start, and the sibling-path seed with coefficient one recovers the witness prefix. The temporary record therefore needs neither a start identifier nor the signature recovery data used by the collision index. The public MQ instance is used to search the remaining d bits.

5.3.1 Cost model

Theorem 10 (Memory and operation counts)**.** *Let $a = h + f$ be the number of bits per endpoint, let b be the number of bits per temporary record, and let R be the RAM budget in bits. Store M endpoints and allocate space for $B = kM$ temporary records, where k is an integer. In one execution, let ℓ_i be the walk length for usable target i, let z_i indicate whether its tag matches, and let r_i be the replay length selected for that target. Set $Z = \sum_i z_i$ over the U usable targets. If one reconstruction pass costs S root evaluations, then the operation counts are*

$$\begin{aligned} R &= M(a + bk), & T_{\text{walk}} &= \sum_i \ell_i, & N_{\text{pass}} &= \left\lceil \frac{Z}{B} \right\rceil, \\ T_{\text{scan}} &= SN_{\text{pass}}, & T_{\text{replay}} &= \sum_i z_i r_i, & T_{\text{val}} &= Z. \end{aligned} \tag{35}$$

The total number of evaluations of the root function is $T_{\text{root}} = T_{\text{walk}} + T_{\text{scan}} + T_{\text{replay}} + T_{\text{val}}$. If Y tag hits survive a v-bit fingerprint, exhaustive d-bit completion uses exactly $Y2^d$ public MQ checks.

For the large-instance estimate, let $\theta \in [C/2^\lambda, 1]$ be the probability that a usable target reaches a retained endpoint after the distance check and before the f-bit tag is tested, and set

$$\alpha = C/2^\lambda, \qquad \zeta = \theta + (1 - \theta)2^{-f}.$$

$L = C/M$, $u = 1/2 - 2^{-\tau}$, $q = \mathbb{E}[U] = Qu$, and $S = gC$. Coverage gives probability α, so $\theta \geq \alpha$; the concrete functional graph may create further endpoint candidates. A true endpoint candidate matches the tag, while any other candidate matches independently with probability 2^{-f}. Under uniform independent tags and fingerprints, and with estimated mean walk and replay length L,

$$\begin{aligned} \mathbb{E}T_{\text{walk}} &= qL, & \mathbb{E}N_{\text{pass}} &\leq \frac{q\zeta}{kM} + 1, & & \\ \mathbb{E}T_{\text{replay}} &= q\zeta L, & \mathbb{E}T_{\text{val}} &= q\zeta, & \mathbb{E}N_{\text{MQ}} &= q\left(\theta + (1 - \theta)2^{-f-v}\right)2^d. \end{aligned} \tag{36}$$

If the final partial pass, replay, and validation terms are omitted while optimizing the continuous memory split, the unique minimizer and its leading cost are

$$\begin{aligned} k^\star &= \sqrt{\frac{ga\zeta}{b}}, \\ T^\star_{\text{lead}} &= \frac{qC}{R}\left(\sqrt{a} + \sqrt{gb\zeta}\right)^2. \end{aligned} \tag{37}$$

Proof. The first identities count the operations performed by the algorithm. In the large-instance estimate, a usable target produces a true endpoint candidate with probability θ; otherwise an independent false tag matches with probability 2^{-f}. Hence $\mathbb{E}Z = q\zeta$, and $\lceil Z/B \rceil \leq Z/B + 1$ proves the conservative pass bound. A seed covered by the table and any incidental true endpoint candidate survive the target fingerprint, whereas a noncandidate survives both filters with probability 2^{-f-v}, proving the final expectation. The mean-length projection gives the remaining terms. Substituting $M = R/(a + bk)$ and $S = gC$ into walk plus triggered-scan work gives

$$\frac{qC}{R}\left(a + bk + g\zeta\left(\frac{a}{k} + b\right)\right).$$

Its second derivative is positive, and setting the first derivative $b - ga\zeta/k^2$ to zero proves Equation (37). □

Equation (35) counts a realized execution. Equation (36) estimates a large instance and upper-bounds the number of batching passes while charging reconstruction cost S. For this estimate, chains are capped at κL. We use the merge load ρ_{cap} below, the geometric termination probability $p_{\mathrm{term}} = 1 - e^{-\kappa}$, and solve

$$g = \frac{\kappa}{p_{\mathrm{term}}(1 - \rho_{\mathrm{cap}}(g))} \tag{38}$$

together with Equation (37) for every integer f. Solutions with $\rho_{\mathrm{cap}} \geq 1$ are discarded. Public MQ checks are counted separately from evaluations of the root function.

Theorem 11 (Signatures and precomputation)**.** *Fix the endpoint layout, tag width, and reconstruction multiplier g. Let C be the measured coverage, write $u = 1/2 - 2^{-\tau}$, and hold the product $P = QC$ fixed. For choices with $Q = P/C \in \mathbb{N}$, take $\theta = \alpha = C/2^{\lambda}$ as the baseline candidate probability. If a larger value of θ is measured, it is substituted directly in Equation (36). Then:*

(i) the success probability is $1 - (1 - uC/2^{\lambda})^{P/C}$ and is nondecreasing in C for fixed P;

(ii) with M retained endpoints and forecast mean $L = C/M$, the forecast online walk term $W = QuL$ equals uP/M, hence is invariant under $C \leftarrow 2^{s}C$ and $Q \leftarrow 2^{-s}Q$;

(iii) when $C/2^{\lambda} \ll 2^{-f}$, the false-tag leading terms for triggered scans and selected replays are invariant, while the estimate also includes the terms for seeds covered by the table from Equation (36); false-tag candidates decrease with Q, whereas preprocessing, the final partial scan, and per-hit reconstruction grow with C;

(iv) for fixed success p and $C/2^{\lambda} = o(1)$,

$$P = (1 + o(1))\frac{-\log(1-p)\,2^{\lambda}}{u}, \qquad W = (1 + o(1))\frac{-\log(1-p)\,2^{\lambda}}{M}.$$

Thus the leading online work is unchanged when Q and C vary with their product fixed.

For chains stopped at the first distinguished point and capped at κL, the random-map estimate for merging is

$$\rho_{\mathrm{cap}} = \frac{M(\kappa L)^2}{2^{\lambda}} = \frac{\kappa^2 C^2}{M 2^{\lambda}}. \tag{39}$$

Correctness and success are conditioned on measured coverage C; Equation (39) estimates the number of chains retained during preprocessing.

Proof. Substitute $Q = P/C$ in Equation (32). The function $\log(1 - x)/x$ is decreasing on $(0, 1)$, proving the first claim. Substitution of $L = C/M$ into QuL proves the second. Equation (36) shows that the contributions from false tags to triggered scans and selected replays are fixed multiples of this quantity for fixed g. Contributions from seeds covered by the table enter through $\theta = \alpha = C/2^{\lambda}$. False tag matches are proportional to Q, and the final scan costs gC. Equation (39) follows from a union bound for one capped chain against M chains. Finally, expand $\log(1 - uC/2^{\lambda})$ at zero and solve the success expression for P. □

5.3.2 Category-I parameters

For Category-I GF(2) short parameters, take $Q = 2^{64}$ and measured coverage $C = 2^{65}$. Conditional on this coverage, the success probability is about 0.631941. Table 4 uses three MPHF bits, a 13-bit endpoint tag, a 256-bit transient record whose required minimum is 232 bits, and a 32-bit target fingerprint, with baseline $\theta = \alpha$. The online and baseline columns report base-two logarithms of the number of evaluations of the root function. The

serial column is $\log_2(Q + T_{\mathrm{pre}} + T_{\mathrm{root}})$; public MQ checks and wall-clock throughput use separate units.

Table 4: Memory and work estimates for the Category-I GF(2) distinguished-point table at $Q = 2^{64}$ and $C = 2^{65}$. The final column is the equal-memory fixed-length baseline.

RAM	$\log M$	$\log L$	online	baseline	gain
2.5 KiB	10.195	54.805	117.931	120.011	2.080
192 MiB	26.462	38.538	101.661	104.011	2.349
52 GiB	34.577	30.423	93.546	96.011	2.465
14 TiB	42.684	22.316	85.439	88.011	2.572
4 PiB	50.877	14.123	77.247	80.011	2.763

These rows all represent 2^{65} distinct signature master seeds. Their one-time construction cost includes starts discarded by the cap and starts lost to merging. Because Equation (1) is shared, the table applies across messages, keys, salts, short and fast variants, and the v2 releases. Each online signature can be processed and discarded immediately.

The 192 MiB and 52 GiB rows replace the ZiB-scale collision index with estimated key-recovery costs of $2^{101.661}$ and $2^{93.546}$ evaluations of the root function. In the 52 GiB row, 47.741 GiB store endpoints and 4.259 GiB store temporary candidates. This division uses the stated engineering estimate for a billion-scale MPHF and preprocessing over 2^{65} states.

Theorem 11 also allows fewer signatures and greater precomputed coverage. The 52 GiB instance uses

$$Q = 2^{52}, \qquad C = 2^{77}, \qquad \log_2 M \approx 34.574601, \qquad \log_2 L \approx 42.425399.$$

The product QC equals 2^{129}, so success conditional on this coverage is about 0.631941. With $\kappa = 4$, the cap gives $\log_2 \rho_{\mathrm{cap}} \approx -4.574601$ and reconstruction multiplier $g \approx 4.253120$. The required 244-bit temporary record fits in its 256-bit allocation. The calculation uses $\theta = \alpha$; a measured rate of additional endpoint candidates is substituted in Equation (36).

The exponents for evaluations of the root function are 93.424695 (target walks), 89.969721 (rescans), 80.424695 (selected replays), and 37.999295 (validation); public MQ checks have exponent 38.021663 in their separate unit because the covered seed passes both filters. The combined online exponent is 93.550759. Adding signatures, preprocessing that includes discarded chains, and online evaluations gives serial exponent 93.550823. Reducing the number of signatures by 2^{12} therefore changes this exponent by 0.0052 bits.

The Q signatures may come from any distribution of public keys because each hit recovers the key belonging to its signature. With one signature per key, the mean number of recovered keys is

$$QuC/2^{128} = 2047/2048,$$

while the probability of at least one recovered key is the value above.

Table 5: Trading signatures for precomputation at fixed success. Each row uses the largest coverage allowed by the displayed merge margin and by a serial overhead of at most 0.1 bit over the corresponding $Q = 2^{64}, C = 2^{65}$ point.

RAM	merge safety	$\log Q$	$\log C$	online	serial
192 MiB	8 bits	58	71	101.661392	101.661392
192 MiB	4 bits	56	73	101.666614	101.666614
52 GiB	8 bits	54	75	93.545905	93.545920
52 GiB	4 bits	52	77	93.550759	93.550823

The reusable table and cross-key collisions can each recover a key. At $B = 3 \cdot 2^{64}$ and $Q = 2^{64}$ with many one-signature Category-I short keys, the mean number of successful table lookups is 6141/4096. Cross-key collisions that recover at least one key have mean $\frac{1}{2}(1 - 2^{-12})$. Their overlap is lower order, so the probability that at least one of these events occurs is

$$1 - \exp\left(-\frac{6141}{4096} - \frac{1}{2}(1 - 2^{-12})\right) = 0.864549 + O(2^{-64}). \tag{40}$$

These two events recover about 2.498779 keys on average: about 1.5 from table hits and one from cross-key collisions. This calculation includes the preprocessing cost and reuses the table across corpora.

For a corpus spread over sufficiently many one-signature keys, keep $B = 3 \cdot 2^{64}$ fixed and increase $\log Q$ from 64 to 65 and 66. The resulting compromise probabilities are 0.864549, 0.993249, and 0.999999, respectively, with expected distinct-key yields 2.498779, 6.996582, and 21.989259. Fewer keys with many signatures reduce the fraction of cross-key pairs but do not change recovery through the single-coordinate table for each individual key.

A pooled search over U users with q signatures each supplies $M_{\text{tar}} \approx Uqp_{\text{tar}}(\tau)$ targets. Its amortized classical cost for one recovered key falls to $2^{\lambda+1}/(Uq)$. NIST lists resistance to multi-key attacks among the desired properties [Nat22]. Binding the fresh salt into root-seed derivation removes both the reusable table and this pooling of targets.

Table 6 collects the collision probabilities after the single-key and cross-key analyses above. The Category-III and Category-V rows refer only to seed collisions within the signing-query limit; exhaustive search and precomputation apply separately in those categories.

Table 6: Classical collision attacks at $Q = 2^{64}$ total signatures. The Category-I values have error $O(2^{-64})$.

Setting	λ	τ	key-recovery probability	result
I, short, one key	128	12	0.393395	one key
I, fast, one key	128	17	0.393467	one key
I, short, many keys	128	12	0.393247	two keys
I, fast, many keys	128	17	0.393462	two keys
III, short/fast	192	18/27	$\approx 2^{-65}$	below the collision bound
V, short/fast	256	25/36	$\approx 2^{-129}$	below the collision bound

6 Root-seed reuse across v2 releases

The attack affects every official v2 tag. In v2.0.0, v2.0.1, v2.0.2, v2.1.0, v2.1.1, and the source revision identified in the artifact, the code passes a zero-initialized `tree_prg_salt` as the salt argument of the PRG call [MQO26b]. The PRG implementation changed between releases, so source-level agreement alone does not establish that a single precomputed table can be reused. We therefore compare the root functions at the block level. In each listed source revision, block j is

$$\mathsf{Enc}_{\mathsf{TweakSalt}(0^\lambda,3,0,j)}(m) \oplus \Psi(m).$$

Later edits add a key-schedule cache, memory-efficient generation, and vectorized paths without changing this scalar map. We compile each revision with its own Makefile and evaluate a common signature master seed in Categories I, III, and V. All builds produce the same first three GGM root seeds byte for byte. A single triangle table therefore applies from v2.0.0 through v2.1.1 and to the listed source revision. The executable test matrix records each commit and fails on any mismatch.

MQOM v3.0 changes this derivation. Its changelog identifies the salt-less root-seed expansion in v2 and states that the issue has been fixed [BBFR26]. The v3 construction is described in Section 9; the attacks in this paper target the v2 releases listed above.

Multi-instance analyses of MPC-in-the-Head signatures require salt binding in GGM puncturable pseudorandom functions to prevent collision attacks [BCC+25]. Ward Beullens reported a separate multi-key issue caused by a short seed in key generation. Matthieu Rivain acknowledged that issue on behalf of the MQOM and SDitH teams and announced a longer KeyGen master seed [Beu24]. The announced change did not alter the signature master seed or its zero-salt root-seed derivation.

7 Origin of the zero-salt call and proof assumptions

The v2.1 discussion of symmetric primitives states that seed derivation, commitment, and expansion incorporate the random signature salt because λ-bit seeds otherwise create collision and accelerated-search risks [BBFR25]. The lower tree levels follow this rule. Algorithm 10 does not: it derives the root-seed array with zero salt.

MQOM v1.0 sampled a signature master seed and derived the root seeds as

$$(r_0, \ldots, r_{\tau-1}) \leftarrow \mathsf{XOF}(\mathsf{salt}, m),$$

so the signature salt entered the root-seed array [FR23]. Version 2.1 replaced this call with Equation (1), introducing the shared root functions used by the attacks above.

The correlated-tree analysis of Feneuil and Rivain invokes the line commitment independently in each of the τ repetitions in Theorem 5 and Figure 6 (Game 1). Its abort analysis counts fresh randomness over a space containing one master salt and τ master seeds: $2^{|\mathsf{msalt}|}(2^{\lambda})^{\tau}$ [FR26]. The v2.1 signer instead samples one λ-bit value m and computes the deterministic vector $(H_0(m), \ldots, H_{\tau-1}(m))$.

Section 5.3 discusses the common batching challenge and grinding as the two main differences in MQOM and states that both preserve the theorem. It does not include the correlation created by one signature master seed. Applying the independent-seed theorem to v2.1 therefore needs another hybrid and a loss for the correlation used in Theorem 1. At minimum, that loss must include collisions in the λ-bit signature master seed. Treating the root-seed array as $\tau\lambda$ fresh bits omits the Category-I attack from the game sequence.

8 Implementation validation and reproducibility

The dependency-free tests enumerate small spaces of coefficient vectors, collisions, ranks, functional graphs, and record allocations. They reconstruct witnesses, reject decoys, regenerate the tables, produce every component of the ledger in Theorem 10, check both record layouts, and solve Equation (38). Their output reports each cost component and keeps public MQ checks separate from evaluations of the root function.

Separate probes run against specified commits of the official C implementation. They execute key generation, signing, verification, challenge parsing, GGM-tree and leaf-seed expansion, witness checking, and secret-key serialization. A v2.1.0 probe forces a collision in mseed and recovers the resulting key. A public 16-bit instance against the current source enumerates the seed from one indexed edge and verifies a fresh forgery. A rank-two probe recovers two keys and verifies one fresh forgery under each key.

The executed instances cover all three security categories, all fields, both speed tradeoffs, and both PIOP round counts. The six-revision matrix checks that a common signature master seed produces the same first three GGM root seeds from v2.0.0 through the current revision. These forced-collision and reduced-domain experiments validate the attack path. The 2^{52}–2^{77} results are analytic evaluations of the large-instance model. The README

lists the commands and commit identifiers for tests, cost regeneration, official probes, and paper compilation.

The probability calculations use uniform signature master seeds and independent Fiat–Shamir challenges in the random-oracle model. The algebraic and combinatorial results are evaluated in that experiment. The implementation results come from finite executable tests against specified source revisions. For the larger tables built with distinguished points, the random-map, fingerprint, MPHF, and engineering parameters are estimates described in Section 5.3. Coverage and operation counts are reported separately.

9 Repair

MQOM v3.0 implements the repair in its correlated-tree variant by deriving, for each execution e, the first top node in Algorithm 21 as

$$\begin{aligned} \widetilde{\mathsf{salt}}_{s,e} &= \mathsf{TweakSalt}\big(\mathsf{salt}_s, 2, \mathsf{IndexIdentifier}(e,0)\big), \\ \mathsf{tree}_s[e][2] = \mathsf{SeedDerive}(\widetilde{\mathsf{salt}}_{s,e}, m_s), &\qquad \mathsf{tree}_s[e][3] = \mathsf{tree}_s[e][2] \oplus \delta. \end{aligned} \tag{41}$$

Root-seed derivation therefore depends on both the signature salt and the execution index. Equal values of m_s under different salts no longer produce the same top nodes, so the shared functions H_e used in Section 5 are absent. The v3.0 changelog explicitly lists the salt-less root-seed expansion identified in this work among the corrected domain-separation issues [BBFR26].

For a fixed execution, input reuse after this change requires the same 2λ-bit pair (salt, m). At $\lambda = 128$ and $Q = 2^{64}$, the mean number of such repeated pairs is about 2^{-129}. Distinct inputs may still collide at the output of the idealized seed-derivation function, but a collision of all τ top nodes has probability $2^{-\tau\lambda}$ per pair in the random-function model.

The repair preserves Equation (2) within each signature: the first top node has changed, while the second is still its XOR with δ. The v3.0 specification, Round-3 implementation package, and updated known-answer tests were published on August 31, 2026 [BBFR26, MQO26a].

10 Conclusion

MQOM v2 derives its GGM root-seed array from the signature master seed without using the signature salt, so two signatures that repeat mseed also repeat every $\mathsf{rseed}[e]$. Since the top seed in each sibling path is either $\mathsf{rseed}[e]$ or $\mathsf{rseed}[e] \oplus \mathsf{FirstBits}_\lambda(x)$, comparing two such signatures can cancel a repeated root seed and reveal $\mathsf{FirstBits}_\lambda(x)$. Three stored values per signature locate every collision that exposes this prefix, after which the attacker reconstructs a GGM tree, recovers the secret MQ solution, and signs a fresh message. At the Category-I limit of 2^{64} signing queries, this passive attack succeeds with probability about 0.39.

The same omitted salt allows root-seed computations to be reused across public keys, parameter sets, and releases, giving the cross-key and precomputation attacks developed from the main result. Tests against the official implementation recover the signing keys and verify fresh forgeries. Root-seed derivation from both the signature master seed and the signature salt, with domain separation by execution, removes the shared relation used by these attacks. MQOM v3.0 implements this repair in its correlated-tree variant and identifies the v2 salt-less root-seed expansion in its changelog.

Acknowledgments

Huge thanks to Thibauld Feneuil and the MQOM team for their careful analysis and scrutiny. They confirmed the salt issue and informed me that the Round-3 version would incorporate the repair without a performance impact. The repair is included in MQOM v3.0.

References

[ABC+26] Gorjan Alagic, Maxime Bros, Pierre Ciadoux, Quynh Dang, Thinh Hung Dang, John Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Hamilton Silberg, Daniel Smith-Tone, and Noah Waller. Status report on the second round of the additional digital signature schemes for the NIST post-quantum cryptography standardization process. Technical Report NIST IR 8610, National Institute of Standards and Technology, 2026. doi:10.6028/NIST.IR.8610.

[BBFR25] Ryad Benadjila, Charles Bouillaguet, Thibauld Feneuil, and Matthieu Rivain. MQOM: MQ on my mind: Algorithm specifications and supporting documentation (version 2.1), 2025. NIST Additional Digital Signatures Round 2 specification, September 22, 2025. URL: https://mqom.org/docs/mqom-v2.1.pdf.

[BBFR26] Ryad Benadjila, Charles Bouillaguet, Thibauld Feneuil, and Matthieu Rivain. MQOM: MQ on my mind: Algorithm specifications and supporting documentation (version 3.0), 2026. NIST Additional Digital Signatures Round 3 specification, August 31, 2026. URL: https://mqom.org/docs/mqom-v3.0.pdf.

[BCC+25] Dung Bui, Eliana Carozza, Geoffroy Couteau, Dahmun Goudarzi, and Antoine Joux. Faster signatures from MPC-in-the-Head. In *Advances in Cryptology—ASIACRYPT 2024*, volume 15484 of *Lecture Notes in Computer Science*, pages 396–428, Singapore, 2025. Springer. doi:10.1007/978-981-96-0875-1_13.

[Beu24] Ward Beullens. Trivial multi-key attacks + attack on ALTEQ. NIST pqc-forum, 2024. Response by Matthieu Rivain on behalf of the SDitH and MQOM teams, February 23, 2024. URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/tjhrrmv837w/m/sjxHooYgBAAJ.

[BFR24] Ryad Benadjila, Thibauld Feneuil, and Matthieu Rivain. MQ on my Mind: Post-quantum signatures from the non-structured multivariate quadratic problem. In *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)*, pages 468–485. IEEE, 2024. doi:10.1109/EuroSP60621.2024.00032.

[Din14] Itai Dinur. Cryptanalytic time-memory-data tradeoffs for FX-constructions with applications to PRINCE and PRIDE. Cryptology ePrint Archive, Paper 2014/656, 2014. URL: https://eprint.iacr.org/2014/656.

[FR23] Thibauld Feneuil and Matthieu Rivain. MQOM: MQ on my mind: Algorithm specifications and supporting documentation (version 1.0), 2023. NIST Additional Digital Signatures submission specification, May 31, 2023. URL: https://mqom.org/docs/mqom-v1.0.pdf.

[FR26] Thibauld Feneuil and Matthieu Rivain. On the security of MPC-in-the-Head signatures with correlated GGM trees. Cryptology ePrint Archive, Paper 2026/615, 2026. URL: https://eprint.iacr.org/2026/615.

[Hel80] Martin E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Transactions on Information Theory*, 26(4):401–406, 1980. doi:10.1109/TIT.1980.1056220.

[LH12] Ga Won Lee and Jin Hong. A comparison of perfect table cryptanalytic tradeoff algorithms. Cryptology ePrint Archive, Paper 2012/540, 2012. URL: https://eprint.iacr.org/2012/540.

[MQO26a] MQOM Team. MQOM resources, 2026. Accessed September 16, 2026; latest listed specification: version 3.0. URL: https://mqom.org/resources.html.

[MQO26b] MQOM Team. MQOM v2 digital signature scheme: Official implementation, 2026. Tags v2.0.0–v2.1.1 and pinned revision 9a5a4d5137c0c9e7974eb8879aa7f13fa6c8ab4d. URL: https://github.com/mqom/mqom-v2.

[Nat22] National Institute of Standards and Technology. Call for additional digital signature schemes for the post-quantum cryptography standardization process, 2022. Updated October 2022. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf.

[Nat26] National Institute of Standards and Technology. Post-quantum cryptography: FAQs, 2026. Updated June 16, 2026; Evaluation Criteria. URL: https://csrc.nist.gov/Projects/post-quantum-cryptography/faqs.