Poly2MsgLeaks: Certified Key Recovery from Polynomial-to-Message Leakage

Yongseong Park[1], YoungChang Kim[1], Jeongdo Kim[2] and Young-Sik Kim[1]

[1] Department of Electrical Engineering and Computer Science, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, Korea
reo91004@dgist.ac.kr, yckim0518@dgist.ac.kr, ysk@dgist.ac.kr

[2] Department of Interdisciplinary Studies of Artificial Intelligence, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, Korea
jd_noahkim@dgist.ac.kr

Abstract. We present Poly2MsgLeaks, a profiled chosen-ciphertext power-analysis attack that recovers the complete 512-coefficient public-key encryption (PKE) secret of SMAUG-T in its TiMER parameter set. Each polynomial-to-message (P2M) decision depends on four ternary secret coefficients and therefore on one of 81 local states. We characterize all admissible P2M predicates and show that our six-query schedule maps the 81 states to 57 binary transcripts; even with both exact polynomial weights, noiseless transcripts leave a mean residual ambiguity of 27.2 bits. Poly2MsgLeaks retains waveform scores for all 81 states, couples the 128 local blocks through the exact weights, and enumerates a calibrated cost region against the public-key relation. Under a uniform-matrix model, we prove that an incorrect key is released with probability at most $2^{-2095.6}$; for exchangeable calibration and evaluation keys, the probability that the true key lies outside the searched region is bounded without a channel model. Using six physical P2M traces recorded on an STM32F415, Poly2MsgLeaks recovers 999 of 1,000 previously unseen secrets, and a verifier-driven policy recovers all 1,000 secrets of a second cohort at a mean cost of 7.021 traces, without false acceptances. With the same decoder and limits, equalizing the scores within each transcript class reduces recovery from 998 to 112 keys. Against single-write message assembly and a two-share masked P2M, the evaluated six-execution attack recovers no complete secret.

Keywords: Side-channel analysis · post-quantum cryptography · SMAUG-T · chosen ciphertext · certified key recovery

1 Introduction

Chosen-ciphertext side-channel attacks turn implementation leakage into an oracle for secret recovery [RSRCB20, RBSRC22, UXT+21]. Chosen ciphertexts can also magnify the leakage of operations that process the secret [XPSR+22]. In lattice-based key encapsulation mechanisms (KEMs), the conversion of a decrypted polynomial into message bits is an attractive target because chosen ciphertexts control its inputs. We study this polynomial-to-message (P2M) step in the TiMER parameter set of SMAUG-T [CCHY24, CCC+24a, The26]. With constant chosen ciphertext polynomials, each of the 128 iterations of the TiMER decoding loop evaluates the same binary predicate on a local state formed by four ternary secret coefficients. A single trace therefore observes 128 local decisions that jointly cover the 512-coefficient secret.

The fundamental challenge stems from the limited information revealed by a small number of traces. Seven binary queries are necessary to separate the 81 local states, and our

six-query schedule maps them to 57 transcripts. Even with both exact polynomial weights, the noiseless transcripts are consistent with a geometric mean of $2^{27.2}$ secrets. The public-key relation identifies the true secret among them with overwhelming probability. This count quantifies residual ambiguity rather than search cost; the practical difficulty is to find the true secret from noisy measurements within a computational budget. Poly2MsgLeaks retains a measured score for each of the 81 states at every block, combines the scores through the exact weights, and enumerates a calibrated cost region against the public relation. A key is released only after this relation and two deterministic KEM checks certify it.

Using six physical P2M traces recorded on an STM32F415 target, a fixed six-query policy recovers 999 of 1,000 secrets generated independently of all profiling and calibration keys, and a verifier-driven policy recovers all 1,000 secrets of a second such cohort at a mean cost of 7.021 traces, without releasing an incorrect key. Equalizing the scores inside each colliding transcript class, while preserving each class's summed state-likelihood score, reduces recovery from 998 to 112 keys with the same inputs, decoder, and limits. We also evaluate the single-write assembly proposed by Kim and Kim [KK26] and a two-share masked P2M.

Contributions. We write Q_6 for our six-query schedule, Q_7 for a fixed seven-query prefix, and Q_{10} for its fixed ten-query extension.

- *Exact predicate geometry and ambiguity.* We characterize the admissible TiMER P2M predicates (60,898 distinct truth tables on 81 states), determine the transcript classes of Q_6, give a lower bound and an exact counting method for the number of secrets consistent with noiseless transcripts and exact weights, and construct a three-query extension that is minimal for Q_7 and the distance-two objective.
- *A certified decoder with complete proofs.* We prove that exact-weight minimum-cost decoding is maximum a posteriori under the fitted model, that our bounded meet-in-the-middle enumeration is complete, and that a public filter analyzed through the chain-ring structure of the TiMER ring is sound; calibration by the maximum gap, an application of split conformal prediction [AB23], bounds the miscoverage rate. These results bound false certification without a channel model.
- *Physical evaluation.* We evaluate two cohorts of 1,000 keys, a score intervention, a solver comparison, and a comparison of extension, repetition, and additional computation, and we apply the same attack to single-write assembly and a two-share masked P2M.

2 Background and Related Work

SMAUG-T and its P2M step. SMAUG-T combines module learning-with-errors key generation with module learning-with-rounding encryption and uses sparse ternary secrets [CCHY24, CCC+24a, CCSS24]. Its KEM applies the Fujisaki–Okamoto (FO) transform with implicit rejection [FO99, HHK17, CCC+24a]. TiMER encodes each message bit into two ring coefficients and decodes with TiMER's two-dimensional adaptation of NewHope-style reconciliation [ADPS16, CCC+24a]: it centers the two coefficients, adds their absolute values, and compares the sum with a threshold. Side-channel analyses of SMAUG-T have addressed P2M message assembly [KK26], the polynomial multiplication [YHH25], and machine-learning-based power analysis [RSK+26]. Kim and Kim identify cumulative byte-wise message assembly as a strong leakage source, recover message bytes from profiled traces, and propose single-write assembly [KK26]. In contrast, we evaluate recovery of the complete 512-coefficient secret with public-key verification.

Chosen-ciphertext attacks. Single-trace attacks exploit number-theoretic transform (NTT) arithmetic [PPM17, PP19] and message encoding [SKL+20, ACLZ20]. Plaintext-checking oracles arise from message decoding, modular reduction, and FO re-encryption [RSRCB20, SPH22, UXT+21]. Masked implementations have been attacked through message-decoding and inverse-NTT leakage [NDGJ21, HHP+21]. Oracle query costs have been reduced systematically [QCZ+21], through multiple-valued outputs [TUX+23, LCS+25], and through parallel queries [RRD+23, GM23]. Shao et al. construct an analytical $9^2 = 81$-state Saber oracle with at most seven cuts and later study optimal ML-KEM query complexity [SLZ24, SLZ+27]. Makaram et al. select Kyber ciphertexts with a satisfiability modulo theories solver [MRD+26]. Berzati et al. recover the first ten ML-KEM-512 known-answer-test keys with six attack traces after $42 + 42$ setup traces [BCVCV26], and Qian et al. combine inverse-NTT and decoding leakage in a one-time chosen-ciphertext attack [QGZ+26].

Soft decoding, side information, and verification. Soft analytical and code-based decoders combine noisy local evidence [VCGS14, GNNJ23, SCZ+22, GNJ26], while sequential testing handles noisy plaintext-checking oracles [PRI26]; side-information frameworks connect it to lattice secrets [DSDGR20, HSMP25], and rank estimation and enumeration quantify and explore the remaining candidate space [VCGS13, VCGRS13]. Collision constraints combined with analog rankings reduce residual ambiguity [BK12]. Meet-in-the-middle steps appear in hybrid lattice attacks [HG07, Wun19, May21], and fixed prefixes with predetermined extensions relate to rate-compatible coding and incremental redundancy [Hag88, Cha85, CT01]. Probabilistic key inference from soft side information is established, for example through belief propagation [VCGS14, QWC+24] and distribution hints [HSMP25]; we do not claim soft inference itself as a contribution. Our contribution is its combination with the exact predicate structure of the TiMER P2M step, the exact secret weights, and the public relation into a certified complete-secret decoder, together with an evaluation at the complete-secret endpoint.

3 Target, Adversary, and Assumptions

Target. We follow the evaluated reference implementation [The26]. TiMER uses $n = 256$, module rank $k = 2$, moduli $q = 1024$, $p = 256$, and $p' = 8$, a 608-byte ciphertext, and a 128-bit message encoded into 256 coefficients. The PKE secret is $s = (s_0, s_1)$ with $s_0, s_1 \in \{-1, 0, 1\}^{256}$, and each polynomial has exactly 70 nonzero coefficients. The public key is (A, b) with $b = -A^T s + e \bmod 1024$, where each error coefficient lies in $[-3, 3]$ and takes the values 0, each sign of 1, 2, and 3 with probabilities 191/512, 247/1024, 67/1024, and 7/1024, respectively.[1] Decryption forms 256 decoder words from c_2 and the module products of c_1 with s. We call the 128 iterations that map word pairs $(i, i+128)$ to message bits the *decode loop*, and the module products together with the decode loop the *P2M body*.

Adversary. The adversary submits chosen ciphertexts to a device that holds a fixed key and records one power trace per execution; it may submit the same ciphertext repeatedly. Because the KEM applies implicit rejection, outputs for invalid ciphertexts are pseudorandom, and the attack uses only the power traces. The public key and the parameter set are known. Profiling uses independently generated keys on the same device and acquisition setup. The goal is exact recovery of s. Our experiments record P2M traces using measurement firmware (Section 6.1) rather than full decapsulation; applying the measured attack to the KEM requires assumption (A5) below.

[1] Version 3.0 of the specification [CCC+24b] lists different TiMER secret-weight and noise parameters; all statements in this paper refer to the distributions of the evaluated implementation [The26].

Assumptions. Our statements rely on the following assumptions; each result names the ones it uses.

(A1) *Key distribution.* The polynomials s_0 and s_1 are independent; each has a uniformly random support of size 70 and independent uniform signs.

(A2) *Uniform matrix.* The matrix A is uniform over $R^{2\times 2}$, with $R = \mathbb{Z}_{1024}[x]/(x^{256}+1)$, and independent of (s, e). This models the deterministic expansion of A from a seed as an ideal random matrix.

(A3) *Channel model.* Given the local states, the block waveforms are independent and Gaussian with state-dependent means and a common covariance.

(A4) *Exchangeability.* Calibration keys and an evaluation key are independent and identically distributed, their traces are acquired under identical conditions, and the schedule, feature extraction, model, integer-conversion rule, and search procedure are fixed independently of the calibration keys; only the threshold τ of Section 5.3 is computed from them.

(A5) *Leakage transfer.* During decapsulation of the encoded ciphertexts, the corresponding P2M computation leaks as in the measurement firmware.

Assumption (A3) is used only for maximum a posteriori optimality (Proposition 7). Soundness of certification uses (A2), coverage uses (A4) or its relaxations in Section 5.3, and applying the measured attack to the KEM uses (A5). Assumption (A1) enters only the probabilistic statements of Section 4 and Proposition 7; certification and coverage require only that each polynomial of the secret has exactly 70 nonzero coefficients and, for coverage, that calibration and evaluation keys share one distribution.

Use of ground truth. The recovery procedure receives the measured traces, the public key, the public weights, and a KEM interface. Ground truth is used only by the acquisition controller to validate captures (Section 6) and to score released keys after all recovery decisions have terminated. Reported recovery rates are conditional on eligible captures.

4 The P2M Oracle and Its Predicate Algebra

4.1 Local states and admissible predicates

At block $i \in \{0, \dots, 127\}$, our routing collects the local state

$$z_i = (s_{0,i}, s_{1,i}, s_{0,i+128}, s_{1,i+128}) \in \mathcal{Z} = \{-1, 0, 1\}^4, \qquad |\mathcal{Z}| = 81. \tag{1}$$

The 128 states partition the 512 secret coefficients. For $z \in \mathcal{Z}$, let $h_\ell(z)$ count the nonzero entries among the two coordinates of z that belong to s_ℓ. Every valid secret satisfies

$$\textstyle\sum_{i=0}^{127} h_0(z_i) = 70, \qquad \sum_{i=0}^{127} h_1(z_i) = 70, \tag{2}$$

and we write $\mathcal{V}$ for the set of state assignments that satisfy (2). An *admissible* query descriptor and its domains are

$$q = (A_0, A_1, c_{2,0}, c_{2,1}), \quad A_0, A_1 \in \{1, \dots, 255\}, \quad c_{2,0}, c_{2,1} \in \{0, \dots, 7\}. \tag{3}$$

Encoding a descriptor at a fixed routing offset yields a 608-byte ciphertext. For $z = (a, b, c, d)$, the module products and the decoder words at a block are

$$\begin{aligned} e_0 &= A_0 a + A_1 b, & x_0 &= (8192 c_{2,0} + 256 e_0) \bmod 65536, \\ e_1 &= A_0 c + A_1 d, & x_1 &= (8192 c_{2,1} + 256 e_1) \bmod 65536. \end{aligned} \tag{4}$$

With $r(x) = \mathrm{int16}(x - 32767)$ and $\mathrm{fa}(x) = (r(x) + (r(x) \gg 15)) \oplus (r(x) \gg 15)$, the reference implementation returns the bit

$$g_q(z) = \Big((\mathrm{fa}(x_0) + \mathrm{fa}(x_1) - 32767) \bmod 65536 \Big) \gg 15. \tag{5}$$

A schedule $Q = (q_1, \ldots, q_m)$ assigns state z the transcript $c_Q(z) = (g_{q_1}(z), \ldots, g_{q_m}(z))$. We write d_Q for the Hamming distance of transcripts and $d_{\min}(Q) = \min_{z \neq z'} d_Q(z, z')$.

Proposition 1 (Reachable predicate structure)**.** *For every admissible q, the words x_0, x_1 are multiples of 256, $\mathrm{fa}(x_\ell) = |x_\ell - 32767| \leq 32767$, and*

$$g_q(z) = 1 \quad \Longleftrightarrow \quad \mathrm{fa}(x_0) + \mathrm{fa}(x_1) \leq 32766. \tag{6}$$

If the nine values of (a, b) and of (c, d) are ordered by $\mathrm{fa}(x_0)$ and $\mathrm{fa}(x_1)$, respectively, with ties broken arbitrarily, then the set $\{z : g_q(z) = 1\}$ is a down-set of the resulting 9×9 product order.

Proof. By (4), $x_\ell = 256 \cdot ((32c_{2,\ell} + e_\ell) \bmod 256)$, so $x_\ell \in \{0, 256, \ldots, 65280\}$. Hence $x_\ell - 32767 \in [-32767, 32513]$ lies in the range of a signed 16-bit integer, the conversion int16 is the identity, and $r = r(x_\ell)$ is odd, so $r \neq -32768$. For such r, the arithmetic shift gives $m = r \gg 15 \in \{0, -1\}$ with $m = -1$ exactly when $r < 0$. If $r \geq 0$, then $\mathrm{fa}(x_\ell) = r$. If $r < 0$, then $(r - 1) \oplus (-1) = -(r - 1) - 1 = -r$. Thus $\mathrm{fa}(x_\ell) = |r| \leq 32767$. The sum $\sigma = \mathrm{fa}(x_0) + \mathrm{fa}(x_1)$ lies in $[0, 65534]$, so $\sigma - 32767 \in [-32767, 32767]$. Bit 15 of $(\sigma - 32767) \bmod 65536$ is set exactly when $\sigma - 32767 < 0$, because the nonnegative values do not exceed 32767. This proves (6). If $g_q(a, b, c, d) = 1$ and (a', b'), (c', d') precede (a, b), (c, d) in the two orders, then $\mathrm{fa}(x'_0) + \mathrm{fa}(x'_1) \leq \mathrm{fa}(x_0) + \mathrm{fa}(x_1) \leq 32766$, so $g_q(a', b', c', d') = 1$. □

Fact 1. *Exact evaluation of all $255^2 \cdot 8^2 = 4{,}161{,}600$ descriptors yields 60,898 distinct truth tables on $\mathcal{Z}$, including the two constant ones.*

Appendix B describes this and the other exhaustive computations. The family of Fact 1 is the search domain for all schedules below.

4.2 Six-query transcript classes

Lemma 1. *Every schedule whose transcripts separate all 81 states has length at least seven, and every such schedule of length seven has $d_{\min} = 1$.*

Proof. Separation requires $2^m \geq 81$, hence $m \geq 7$. If a separating schedule of length seven had $d_{\min} \geq 2$, deleting one coordinate would keep all 81 transcripts distinct, which is impossible in $\{0, 1\}^6$. □

Table 1 lists our schedules. The prefix Q_7 separates all states; it was fixed before profiling, because all profiling traces use it, and the results of this section depend only on its listed descriptors. The standalone schedule removes its second coordinate,

$$Q_6 = (q_1, q_3, q_4, q_5, q_6, q_7), \tag{7}$$

and was selected with the calibration keys. We screened all $\binom{10}{6} = 210$ six-query subsets on 100 alternating entries of the sorted calibration roster using likelihood and exact-weight proxies. Five candidate schedules were compared by two-fold cross-calibrated exact decoding on all 200 calibration keys; Q_6 uniquely attained the highest recovery rate, 195/200, and the pooled calibration set then determined τ_6. Section 5.3 accounts for this data-dependent selection. For a transcript w of Q_6, we call $F_w = \{z \in \mathcal{Z} : c_{Q_6}(z) = w\}$ its *transcript class*.

Table 1: Descriptors $(A_0, A_1, c_{2,0}, c_{2,1})$ of the fixed schedules $Q_7 = (q_1, \ldots, q_7)$, $Q_6 = Q_7 \setminus \{q_2\}$, and $Q_{10} = Q_7 \,\|\, (q_8, q_9, q_{10})$.

Query	Tuple	Query	Tuple	Query	Tuple
q_1	$(3, 94, 5, 7)$	q_5	$(131, 255, 2, 2)$	q_9	$(22, 108, 7, 5)$
q_2	$(66, 193, 7, 1)$	q_6	$(161, 129, 5, 7)$	q_{10}	$(209, 193, 4, 1)$
q_3	$(81, 2, 0, 7)$	q_7	$(161, 130, 3, 1)$		
q_4	$(87, 202, 1, 7)$	q_8	$(66, 63, 6, 2)$		

Proposition 2 (Transcript classes of Q_6)**.** *(a) Q_6 partitions $\mathcal{Z}$ into 33 singleton and 24 two-element transcript classes.*

(b) Exactly four two-element classes are weight-neutral, that is, their two states have equal (h_0, h_1). They are $\{(0, \pm 1, -1, 0)\}$, $\{(0, 0, -1, \pm 1)\}$, $\{(1, \pm 1, 0, 0)\}$, and $\{(1, 0, 0, \pm 1)\}$; in each, the states differ only in the sign of one coefficient of s_1, and $(h_0, h_1) = (1, 1)$.

(c) If a valid secret has K blocks in weight-neutral classes, then at least 2^K valid secrets share its noiseless Q_6 transcripts.

(d) Under (A1), $\mathbb{E}[K] = 47089/4624 \approx 10.18$.

Proof. Parts (a) and (b) follow from evaluating the 81×6 transcript matrix (Appendix B). For (c), each of the K blocks can independently keep its state or switch to the other state of its class. The 2^K resulting assignments are distinct, have the same transcripts, and satisfy (2) because the two states of a weight-neutral class contribute the same (h_0, h_1). For (d), let $S \subseteq \{0, \ldots, 255\}$ be the support of s_0. Under (A1), $\Pr[i \in S] = 70/256$ and $\Pr[i + 128 \notin S \mid i \in S] = 186/255$, so for each sign σ, $\Pr[s_{0,i} = \sigma,\, s_{0,i+128} = 0] = \frac{70}{256} \cdot \frac{186}{255} \cdot \frac{1}{2} = \frac{217}{2176}$, and the same value holds for $(s_{0,i}, s_{0,i+128}) = (0, \sigma)$ and for s_1. Each of the eight weight-neutral states fixes exactly one nonzero coordinate with a given sign in each polynomial, so by independence of s_0 and s_1 it has probability $(217/2176)^2$. A block therefore lies in a weight-neutral class with probability $8(217/2176)^2 = 47089/591872$, and linearity of expectation over 128 blocks gives $\mathbb{E}[K] = 47089/4624$. No independence across blocks is needed. □

The bound in (c) is loose, because swaps in classes with different weight contributions can compensate each other. The number of consistent secrets can be computed exactly. For a schedule Q and a valid secret s, let $N_Q(s)$ denote the number of $s' \in \mathcal{V}$ with $c_Q(z_i(s')) = c_Q(z_i(s))$ for all i, and define the bivariate polynomial $\Phi_w(x, y) = \sum_{z \in F_w} x^{h_0(z)} y^{h_1(z)}$ for each class F_w.

Proposition 3 (Exact count of consistent secrets)**.** *Let $w_i = c_Q(z_i(s))$. Then $N_Q(s)$ equals the coefficient of $x^{70}y^{70}$ in $\prod_{i=0}^{127} \Phi_{w_i}(x, y)$ and can be computed with $O(128 \cdot 71^2 \cdot 9)$ integer operations.*

Proof. The routing (1) is a bijection between secrets and state assignments, so the consistent secrets correspond to tuples $(z_i')_i$ with $z_i' \in F_{w_i}$ that satisfy (2). Expanding the product, each tuple contributes the monomial $x^{\sum_i h_0(z_i')} y^{\sum_i h_1(z_i')}$, so the coefficient of $x^{70}y^{70}$ counts exactly the consistent tuples. Multiplying the factors one at a time and discarding monomials of degree above 70 in either variable, which cannot contribute to $x^{70}y^{70}$, requires at most $71^2 \cdot 9$ multiply-add operations per block. □

Evaluating Proposition 3 on a deterministic sample of 10,000 keys drawn under (A1) gives a sample mean of 27.15 bits for $\log_2 N_{Q_6}$ (95% Monte Carlo confidence interval $[27.03, 27.27]$), with 5th and 95th percentiles 16.87 and 37.23 bits and range $[7.58, 48.91]$. The expected number of blocks in two-element classes is exactly $128 \cdot 23569749/37879808 \approx$

Table 2: Exact transcript geometry over all $\binom{81}{2} = 3{,}240$ state pairs. For Q_6, the pairs at $d_{\min} = 0$ are the 24 colliding pairs. The last column is the mean residual ambiguity under noiseless transcripts and exact weights (Proposition 3).

Schedule	Length	$d_{\min}$	Transcripts	Pairs at $d_{\min}$	$\mathbb{E}[\log_2 N_Q]$
Q_6	6	0	57	24	27.15
Q_7	7	1	81	182	0
R_{10} (repeat q_4, q_5, q_6)	10	1	81	98	0
Q_{10} (append q_8, q_9, q_{10})	10	2	81	195	0

79.64. Among the seven single-coordinate deletions of Q_7, removing $q_1, \ldots, q_7$ leaves 31.6, 26.8, 10.3, 51.5, 49.5, 41.2, and 57.8 bits on average over the first 2,000 keys of the same sample. Ambiguity under noiseless transcripts alone does not determine recovery from measured scores, which depends on how well the waveform separates the states inside each class. These counts quantify residual ambiguity; they are not lower bounds on the cost of recovery, because a decoder can exclude many candidates at once through partial residuals and set-valued joins (Section 5.4).

Corollary 1 (Information in noiseless transcripts)**.** *Assume (A2). For every $s^\star$ and every $e \in [-3, 3]^{512}$, the probability over A that some secret $s \neq s^\star$ consistent with the noiseless Q_6 transcripts of $s^\star$ passes the public filter (15) is at most $2^{-2095.6}$.*

Proof. Consistent secrets satisfy (2), so the event is contained in the event bounded by Theorem 1. □

Noiseless transcripts, the weights, and the public key thus determine the secret. Under measurement noise, however, the transcripts are themselves uncertain, and a decoder that scores candidates only by transcript likelihood assigns equal cost to all members of a consistent set. Section 7.2 measures, for our decoder, the effect of the waveform scores within the classes.

4.3 Fixed redundancy and minimal repair

Proposition 4 (Nearest-pair identity)**.** *Let $S \subseteq \{0,1\}^7$ contain the 81 transcripts of a separating seven-query schedule, and let $e(S)$ count the hypercube edges with both endpoints in S. Then $e(S) = 119 + e(S^c) \geq 119$.*

Proof. Let ∂S be the set of edges between S and its 47-element complement. Each vertex has degree 7, so $7|S| = 2e(S) + |\partial S|$ and $7|S^c| = 2e(S^c) + |\partial S|$. Subtracting gives $2(e(S) - e(S^c)) = 7(81 - 47) = 238$. □

Q_7 has 182 pairs at distance one. Each such pair has a unique separating coordinate, and the coordinates of Q_7 are the separating coordinate for 24, 24, 24, 29, 28, 27, and 26 pairs, respectively. Removing q_2 merges exactly the 24 pairs of its class, which yields the 24 two-element classes of Q_6. Table 2 compares the fixed extension with R_{10}, which repeats q_4, q_5, q_6.

Proposition 5 (Repetition and extension coverage)**.** *Repeating coordinates of a separating seven-query schedule raises $d_{\min}$ to two only after all seven coordinates have been repeated. For Q_7, any three repeated coordinates, with or without multiplicity, leave at least 98 pairs at distance one, whereas q_8, q_9, q_{10} separate all 182 pairs.*

Proof. Repeating coordinate r increases the distance of a nearest pair exactly when r is the separating coordinate of that pair. By Lemma 1, deleting any coordinate merges at least two transcripts, and these transcripts form a nearest pair whose separating coordinate

is the deleted one. Every coordinate therefore separates at least one nearest pair and must be repeated. Three repetitions cover at most the union of three coordinate classes, at most $29 + 28 + 27 = 84$ pairs, which leaves at least $182 - 84 = 98$. Exact evaluation shows that q_8, q_9, and q_{10} separate 114, 105, and 86 of the 182 pairs, with union equal to all 182. □

Proposition 6 (Minimal repair of the fixed prefix)**.** *For Q_7, the predicate family of Fact 1, and the objective $d_{\min} \geq 2$, three additional predicates are necessary and sufficient.*

Proof. Sufficiency follows from Proposition 5. For necessity, an appended predicate raises the distance of a nearest pair exactly when it separates the pair, so we need predicates whose separated sets cover all 182 pairs. Restricting the 60,898 truth tables to these pairs yields 47,684 distinct nonempty sets. The largest contains 114 pairs, so no single predicate suffices, and exhaustive testing of all unordered pairs of these sets finds no union of size 182 (Appendix B). □

5 From Traces to a Certified Key

5.1 Waveform scores and integer costs

A decode-loop trace contains 128 iterations. We align at the trigger, extract the same window from each iteration, and average it into 30 cycle-aligned bins. The measured iteration period is approximately 2,031 samples at 500 MS/s, so with the 7.384615 MHz clock the reduction retains approximately one sample mean per processor cycle. Concatenation over queries gives features in $\mathbb{R}^{180}$, $\mathbb{R}^{210}$, and $\mathbb{R}^{300}$ for Q_6, Q_7, and Q_{10}. Each trace is one unaveraged execution.

Profiling uses two 112-key cohorts. One follows the natural state distribution, and the other contains valid keys selected to increase the support of rare states; its least frequent state has 72 observations. Known secrets label all $224 \cdot 128 = 28{,}672$ blocks. For stage $m \in \{7, 10\}$, let $(X_n^{(m)}, z_n)_{n=1}^N$ be the pooled labeled blocks with $X_n^{(m)} \in \mathbb{R}^{d_m}$. We estimate one mean $\mu_z^{(m)}$ per state and the pooled covariance

$$\Sigma_0^{(m)} = \frac{1}{N - 81} \sum_{n=1}^{N} (X_n^{(m)} - \mu_{z_n}^{(m)})(X_n^{(m)} - \mu_{z_n}^{(m)})^T, \tag{8}$$

which we regularize as $\Sigma^{(m)} = \Sigma_0^{(m)} + \lambda_m I$ with $\lambda_m = 10^{-3} \operatorname{tr}(\Sigma_0^{(m)})/d_m + 10^{-9}$. Let P_6 select the six query blocks of (7) from a Q_7 feature. Gaussian marginalization under this linear selection gives the Q_6 model exactly: $X^{(6)} = P_6 X^{(7)}$, $\mu_z^{(6)} = P_6 \mu_z^{(7)}$, and $\Sigma^{(6)} = P_6 \Sigma^{(7)} P_6^T$. Q_{10} has its own 300-dimensional fit. For a fixed stage, the common-covariance Gaussian score [CRR03, MOP07] of state z at block i is

$$\ell_i(z) = X_i^T \Sigma^{-1} \mu_z - \tfrac{1}{2} \mu_z^T \Sigma^{-1} \mu_z, \tag{9}$$

which equals the Gaussian log-density up to terms that do not depend on z. We define real costs $\gamma_{i,z} = \max_{z'} \ell_i(z') - \ell_i(z) \geq 0$ from the stored binary64 scores. For each key k, let S_k be the least common multiple of their exact rational denominators and use $c_{i,z} = S_k \gamma_{i,z}$. Thus every $c_{i,z}$ is an integer and this conversion introduces no additional rounding. For a valid secret s, let $\Gamma(s) = \sum_i \gamma_{i,z_i(s)}$ and $C(s) = \sum_i c_{i,z_i(s)}$; we suppress the key index on these two quantities and on c below.

5.2 Exact-weight inference

The dynamic program

$$D_0(0,0) = 0, \qquad D_{i+1}(u,v) = \min_{z \in \mathcal{Z}} \{D_i(u - h_0(z), v - h_1(z)) + c_{i,z}\}, \tag{10}$$

with $D_0(u,v) = +\infty$ for $(u,v) \neq (0,0)$, followed by backtracking from $D_{128}(70,70)$, returns $\widehat{s}_{\mathrm{wt}} \in \arg\min_{\mathcal{V}} C$.

Lemma 2. *Let a sampler scan n positions in order and select each position with probability k'/n', where k' positions remain to be selected among the n' remaining positions. Every k-subset is selected with probability $\binom{n}{k}^{-1}$.*

Proof. By induction on n; the cases $k = 0$ and $k = n$ are immediate. A k-subset that contains the first position has probability $\frac{k}{n}\binom{n-1}{k-1}^{-1} = \binom{n}{k}^{-1}$, and one that does not has probability $\frac{n-k}{n}\binom{n-1}{k}^{-1} = \binom{n}{k}^{-1}$. □

The reference implementation generated the fresh calibration and evaluation keys from distinct deterministic seeds. Its fixed-weight sampler selects a uniform 70-position support as in Lemma 2 and independent uniform signs, so (A1) holds for these cohorts. Balanced profiling deliberately conditions supports to cover the local states; it estimates the channel model and is not part of calibration/evaluation exchangeability.

Proposition 7 (Exact-weight maximum a posteriori)**.** *Under (A1) and (A3), every minimizer of Γ over $\mathcal{V}$ is a maximum a posteriori assignment given the observations, and the recurrence (10) with real costs γ computes $\min_{\mathcal{V}} \Gamma$.*

Proof. Under (A1), each valid secret has prior probability $(\binom{256}{70}^{-1}2^{-70})^2$, and every other assignment has probability zero. Under (A3), the likelihood is $\prod_i \mathcal{N}(X_i; \mu_{z_i}, \Sigma)$. Because the covariance is common, the normalizing constant and the term $-\frac{1}{2}X_i^T\Sigma^{-1}X_i$ do not depend on z_i, so $\log\mathcal{N}(X_i;\mu_z,\Sigma) = \ell_i(z) + \kappa_i$ with κ_i independent of z. Hence

$$\Pr[s \mid X] \propto \mathbf{1}[s \in \mathcal{V}] \exp\Big(\sum_i \ell_i(z_i(s))\Big) = \mathbf{1}[s \in \mathcal{V}] \exp\Big(\sum_i \max_{z'} \ell_i(z') - \Gamma(s)\Big), \tag{11}$$

and maximizing the posterior is equivalent to minimizing Γ over $\mathcal{V}$. For the recurrence, induction on i shows that $D_i(u,v)$ is the minimum cost of an assignment to blocks $0,\dots,i-1$ with weight sums (u,v), because the last block of such an assignment takes some state z and the preceding blocks form an assignment with sums $(u - h_0(z), v - h_1(z))$. Thus $D_{128}(70,70) = \min_{\mathcal{V}} \Gamma$. □

Corollary 2 (Integer objective)**.** *For the stored scores of key k, $C(s) = S_k\Gamma(s)$ for every $s \in \mathcal{V}$. Hence C and Γ have exactly the same minimizers over $\mathcal{V}$.*

Proof. The equality follows term by term from $c_{i,z} = S_k\gamma_{i,z}$, and multiplication by the positive constant S_k preserves the ordering. □

All subsequent statements concern the integer objective C and are exact.

5.3 Calibrated region and coverage

For calibration key $h \in \{1,\dots,n\}$ with secret $s_h^\star$ and scale S_h, let $G_h = (C_h(s_h^\star) - C_h(\widehat{s}_{\mathrm{wt},h}))/S_h \geq 0$ be its normalized calibration gap and $\tau = \max_h G_h$. For an evaluation key k, the decoder searches the region

$$\mathcal{R}_k = \{s \in \mathcal{V} : C_k(s) \leq B_k\}, \qquad B_k = C_k(\widehat{s}_{\mathrm{wt},k}) + \lfloor S_k\tau \rfloor. \tag{12}$$

We suppress the key index on B_k and $\mathcal{R}_k$ below. With $n = 200$ calibration keys, the exact normalized thresholds are $1283901629293439/2^{45}$, $489645669612287/2^{44}$, and $524650441082833/2^{45}$ for Q_6, Q_7, and Q_{10}, respectively.

Proposition 8 (Calibration coverage)**.** *Under (A4), the gap G_{n+1} of an evaluation key satisfies* $\Pr[G_{n+1} > \tau] \le 1/(n+1)$*. Hence the true secret lies in $\mathcal{R}$ with probability at least $n/(n+1)$, and among N evaluation keys the expected number outside their regions is at most $N/(n+1)$.*

Proof. Under (A4), the gaps $G_1, \dots, G_{n+1}$ are exchangeable. Let $E_j = \{G_j > \max_{i \neq j} G_i\}$. These events are pairwise disjoint, because at most one gap can strictly exceed all others, and they have equal probability by exchangeability. Hence $(n+1)\Pr[E_{n+1}] \le 1$. The event $\{G_{n+1} > \tau\}$ equals E_{n+1}. The second statement follows by linearity of expectation. □

For $n = 200$ and $N = 1{,}000$, the bound is 4.98 keys. Proposition 8 is the rank argument of split conformal prediction [AB23]. The bound is marginal, that is, averaged over the calibration and evaluation keys; it does not hold conditionally on a fixed calibration set or on a subpopulation selected after observation, such as the keys that reach the Q_{10} stage. Reusing the calibration keys to choose among configurations weakens it as follows.

Proposition 9 (Coverage after selection)**.** *Let configurations $j = 1, \dots, m$ each satisfy (A4), with gaps $G_1^{(j)}, \dots, G_{n+1}^{(j)}$ on a common probability space and thresholds $\tau^{(j)} = \max_{h \le n} G_h^{(j)}$. If an index $\widehat{j}$ is chosen as any function of the calibration data, then* $\Pr[G_{n+1}^{(\widehat{j})} > \tau^{(\widehat{j})}] \le m/(n+1)$.

Proof. The event $\{G_{n+1}^{(\widehat{j})} > \tau^{(\widehat{j})}\}$ is contained in $\bigcup_j \{G_{n+1}^{(j)} > \tau^{(j)}\}$, and each event of the union has probability at most $1/(n+1)$ by Proposition 8. □

In our experiments, Q_7 and Q_{10} were fixed before their profiling traces were acquired, so Proposition 8 applies to τ_7 and τ_{10} provided that no other cost-affecting setting was chosen with the calibration keys. The calibration keys were also used to select Q_6 (Section 4.2); Proposition 9 then applies to τ_6 with m equal to the size of the candidate family, which must be fixed before calibration. Settings that do not change the costs, such as time and work limits, do not count toward m. Here the family contains all $\binom{10}{6} = 210$ schedules, so the bound 210/201 is trivially loose (exceeding 1). The independent evaluation cohort instead provides a held-out check of the selected configuration.

Proposition 10 (Held-out coverage)**.** *Fix the calibration data, and let p be the probability that an evaluation key satisfies $G > \tau$. If the N evaluation keys are independent and identically distributed, independent of the calibration data, and acquired under identical conditions, and at most x of them satisfy $G > \tau$, then $U = \mathrm{Beta}^{-1}(0.95; x+1, N-x)$ satisfies* $\Pr[p \le U] \ge 0.95$.

Proof. Given the calibration data, τ and the selected configuration are fixed, so the indicators $\mathbf{1}[G_k > \tau]$ of the evaluation keys are independent Bernoulli variables with parameter p. The Clopper–Pearson bound [CP34] U is an exact one-sided 95% upper confidence bound for p, and it is nondecreasing in x, so an upper bound on the number of exceedances yields a valid bound. □

Every key certified with its true secret satisfies $G \le \tau$ (Proposition 11), so the counts of Section 7 bound x without further measurements, and the check does not depend on how the configuration was chosen. The check concerns the population of eligible evaluation keys and holds conditionally on the realized calibration data.

5.4 Bounded enumeration

Let $\bar{w} = (70, 70)$. For a set U of blocks, a partial assignment $p = (z_i)_{i \in U}$ has cost $C(p) = \sum_{i \in U} c_{i,z_i}$ and weight vector $u(p) = \sum_{i \in U} (h_0(z_i), h_1(z_i))$, and

$$L_U(a,b) = \min\Big\{ C(p) : p \in \mathcal{Z}^U,\ u(p) = (a,b) \Big\} \tag{13}$$

Algorithm 1 DecodeAndCertify with integer costs c, public key pk, normalized threshold τ, scale S_k, and resource limits

1: Partition the blocks into four consecutive 32-block segments $U_1, \dots, U_4$; set $H_L = U_1 \cup U_2$ and $H_R = U_3 \cup U_4$.
2: Set $\Lambda = (0, 1, \dots, \lfloor \tau \rfloor, \tau)$, omitting a repeated endpoint; initialize cumulative work and, for the capped Q_6 stage, an empty cache of checked candidates.
3: For each $\lambda \in \Lambda$ in order, set $B_\lambda = C(\widehat{s}_{\mathrm{wt}}) + \lfloor S_k \lambda \rfloor$ and execute steps 4–7 with the remaining cumulative budget.
4: Form each segment set P_j and the two half sets P_L, P_R using the exact completion-cost tests of (13) at bound B_λ.
5: Index every $p_R \in P_R$ by $(u(p_R), \rho_J(p_R))$, retaining all entries at each index, and join P_L against all targets in T_J.
6: Complete the public enumeration at B_λ and scan all its candidates in nondecreasing cost against both KEM challenges. The capped Q_6 stage skips candidates already in its cache. Stop after all candidates tied at the first successful cost.
7: If a work or candidate limit prevents this decision, return ABSTAIN; if exactly one candidate at the first successful cost passes both challenges, return it as CERTIFIED; if several pass, return ABSTAIN; if there is no success, continue to the next layer.
8: If the final layer completes without a successful candidate, return ABSTAIN.

is its exact completion cost, computed by (10) restricted to U ($+\infty$ if no such p exists). For the public key (A, b), choose two coefficient coordinates j_1, j_2; our implementation uses coefficients 0 and 1 of the first module output. Let

$$\rho_J(s) = ((A^T s)_{j_1}, (A^T s)_{j_2}) \in \mathbb{Z}_{1024}^2, \qquad T_J = \{(-b_{j_1} + \varepsilon_1, -b_{j_2} + \varepsilon_2) \bmod 1024 : \varepsilon \in [-3, 3]^2\}. \tag{14}$$

Here s is viewed as a vector in $\{-1, 0, 1\}^{512}$ through the routing bijection, so ρ_J is linear and $|T_J| = 49$. The public filter accepts the set

$$\mathcal{F}(pk) = \left\{ s \in \mathcal{V} : \mathrm{center}_{1024}(b + A^T s) \in [-3, 3]^{512} \right\}, \tag{15}$$

where center_{1024} maps each residue into $[-512, 511]$.

Lemma 3 (Residual projection)**.** *Every $s \in \mathcal{F}(pk)$ satisfies $\rho_J(s) \in T_J$.*

Proof. For $s \in \mathcal{F}(pk)$, each coordinate satisfies $b_j + (A^T s)_j \equiv \varepsilon_j \pmod{1024}$ with $\varepsilon_j \in [-3, 3]$, so $(A^T s)_j \equiv -b_j + \varepsilon_j$. Restricting to j_1, j_2 gives an element of T_J. □

Lemma 4 (Set-valued join)**.** *Let $H_L \cup H_R$ partition the blocks, and let $s \in \mathcal{F}(pk)$ have restrictions p_L and p_R. If p_R is stored in the list indexed by $(u(p_R), \rho_J(p_R))$, then querying the lists $(\bar{w} - u(p_L), t - \rho_J(p_L))$ for all $t \in T_J$ returns p_R.*

Proof. Weights and ρ_J are additive over the partition, so $u(p_R) = \bar{w} - u(p_L)$ and $\rho_J(p_R) = \rho_J(s) - \rho_J(p_L)$. By Lemma 3, $t = \rho_J(s) \in T_J$, and the query for this t addresses the list that contains p_R. □

Algorithm 1 processes increasing integer gap layers. Segment and half lists are pruned with exact completion costs, the set-valued index preserves every selected public residual, work is accumulated across layers, and the capped Q_6 stage caches KEM-check results. Once the first successful cost is found, the KEM scan resolves every tie at that cost but does not test higher-cost candidates. The Q_6 stage additionally caps the cumulative number of distinct KEM candidate checks at 4,096.

Proposition 11 (Completeness of bounded enumeration)**.** *If Algorithm 1 certifies a candidate of cost $c^\star$ at layer λ, it has tested every candidate in $\mathcal{F}(pk)$ of cost at most $c^\star$ against both KEM challenges. Consequently, the output is the unique minimum-cost*

qualifying element of $\mathcal{R} \cap \mathcal{F}(pk)$. If the algorithm reaches the final layer without a resource limit and this element is unique, it certifies it.

Proof. Let s satisfy $C(s) \le c^\star \le B_\lambda$, with restrictions p_j to U_j, p_L to H_L, and p_R to H_R. The restriction of s to the complement of U_j has weight vector $\bar{w} - u(p_j)$, so $C(p_j) + L_{\overline{U_j}}(\bar{w} - u(p_j)) \le C(s) \le B_\lambda$ and $p_j \in P_j$. The same argument gives $p_L \in P_L$ and $p_R \in P_R$. Lemma 4 then generates s for $t = \rho_J(s)$. The scan proceeds in nondecreasing cost and certification occurred, so every such candidate, including every tie at $c^\star$, was tested. A candidate outside the completed layer has cost greater than $B_\lambda \ge c^\star$ and cannot improve upon or tie the optimal candidate. At the final layer, $B_\lambda = B$, which proves the last statement. □

Without the 49-element target set, the join would be lossy: a single target $-b_J$ retains the true residual only when both selected errors are zero, which under the error distribution of Section 3 has probability $(191/512)^2 \approx 0.139$.

5.5 Soundness of the public filter

The true secret satisfies $b + A^T s^\star = e \in [-3, 3]^{512}$, so $s^\star \in \mathcal{F}(pk)$. The filter rejects every incorrect secret candidate with overwhelming probability.

Lemma 5 (Image size for secret differences)**.** *Let $R = \mathbb{Z}_{1024}[x]/(x^{256} + 1)$. For every nonzero $d = (d_0, d_1) \in R^2$ with coefficients in $\{-2, -1, 0, 1, 2\}$, the map $\phi_d : R^2 \to R$, $(a_0, a_1) \mapsto a_0 d_0 + a_1 d_1$, has an image of size at least 2^{2049}.*

The proof in Appendix A is self-contained and uses the chain-ring structure of R [Din05, DLP04].

Theorem 1 (Soundness of the public filter)**.** *Assume (A2). For every $s^\star$, every $e \in [-3, 3]^{512}$, and every $s \in \mathcal{V} \setminus \{s^\star\}$, $\Pr_A[s \in \mathcal{F}(pk)] \le (7^{256}/2^{2049})^2 < 2^{-2660.6}$. The expected number of such s in $\mathcal{F}(pk)$ is below $2^{-2095.6}$, and so is the probability that one exists.*

Proof. Fix $s^\star$ and e; by (A2), A is uniform and independent of both. Let $d = s - s^\star \ne 0$; its coefficients lie in $\{-2, \dots, 2\}$. Since $b + A^T s = e + A^T d$, the secret s passes the filter exactly when $e_j + (A^T d)_j \in [-3, 3]^{256}$ (after centering) for both module components $j \in \{0, 1\}$, where $(A^T d)_j = \phi_d(A_{0j}, A_{1j})$. The map ϕ_d is a homomorphism of additive groups, so for uniform (A_{0j}, A_{1j}) the value $\phi_d(A_{0j}, A_{1j})$ is uniform on the image I_d, whose size is at least 2^{2049} by Lemma 5. The accepted values form a set of 7^{256} elements, so component j passes with probability at most $7^{256}/2^{2049} < 2^{-1330.3}$. The two columns of A are independent, which gives the first bound. The number of exact-weight secrets is $(\binom{256}{70} 2^{70})^2 < 2^{565.01}$, and this set does not depend on A. Linearity of expectation bounds the expected number of incorrect secrets in $\mathcal{F}(pk)$ by $2^{565.01 - 2660.63} < 2^{-2095.6}$, and Markov's inequality bounds the probability that one exists. □

5.6 Certified release

The two KEM challenges are honest encapsulations derived from public domain-separated inputs; a candidate passes when its decapsulations return the expected 32-byte keys. Let D denote the event that one of these challenges is a decryption failure under $s^\star$, and let X denote the event that Algorithm 1 reaches a resource limit.

Theorem 2 (Certified recovery)**.** *Let $(\widehat{s}, \rho)$ be the output of Algorithm 1 with the calibrated bound (12). Under (A2),*

$$\Pr[\rho = \textsc{Certified} \wedge \widehat{s} \ne s^\star] \le 2^{-2095.6}.$$

Table 3: Fixed parameters of the recovery stages. The table reports each normalized threshold to four decimals.

Stage	Traces	Feature dimension	Normalized τ	Work bound	Candidate checks
Q_6	6	180	36.4907	5×10^8	4,096
Q_7	7	210	27.8331	5×10^7	–
Q_{10}	10	300	14.9115	5×10^7	–

Under (A2) and (A4),

$$\Pr[\rho = \textsc{Certified} \wedge \widehat{s} = s^\star] \geq 1 - \tfrac{1}{n+1} - \Pr[\mathsf{X}] - \Pr[\mathsf{D}] - 2^{-2095.6}.$$

Neither statement uses the channel model (A3).

Proof. Let U be the event $\mathcal{F}(pk) = \{s^\star\}$; by Theorem 1, $\Pr[\neg\mathsf{U}] < 2^{-2095.6}$. Every certified output lies in $\mathcal{F}(pk)$, so a certified incorrect output implies $\neg\mathsf{U}$, which proves the first statement. For the second, consider the event $\mathsf{U} \cap \{G_{n+1} \leq \tau\} \cap \neg\mathsf{X} \cap \neg\mathsf{D}$. On this event, $s^\star \in \mathcal{R} \cap \mathcal{F}(pk)$ passes both challenges. Because $\mathcal{F}(pk) = \{s^\star\}$, it is the unique qualifying element in the calibrated region; $\neg\mathsf{X}$ lets the decoder reach and complete the required layer, so Proposition 11 certifies it. A union bound with Proposition 8 completes the proof. □

Theorem 2 separates the roles of the components: certification controls false release, calibration bounds the miscoverage probability, and the channel model affects only the resources needed to complete the search. After data-dependent selection, $1/(n+1)$ is replaced by the bound of Proposition 9. The theorem does not bound $\Pr[\mathsf{X}]$; recovery within finite limits is an empirical result (Section 7). Because the first statement holds for arbitrary traces, a failure of (A5) can lower the success rate on the decapsulation path but not raise the probability of releasing an incorrect key.

5.7 Fixed recovery policies

Table 3 lists the values fixed before evaluation; Section 7.4 compares a wider Q_7 threshold, repeated observations, and the fixed extension. The work bound is cumulative within a stage, and the deterministic work counter covers segment expansion, pair generation, selected-residual lookup, and full public checks. The standalone policy runs Algorithm 1 once on Q_6. The verifier-driven policy runs it on Q_7 and, if no key is certified, appends q_8, q_9, q_{10} and runs it on Q_{10} with its own model; with N_{ext} extensions among N keys, its mean cost is $7 + 3N_{\text{ext}}/N$ traces. Both policies were fixed before the evaluation cohorts were acquired, and no evaluation trace changes a model, schedule, threshold, or limit.

6 Experimental Setup

6.1 Target, firmware, and acquisition

We evaluate TiMER on an STM32F415 clocked at 7.384615 MHz. Firmware derives from reference version 1.2.0 and specification 260521 [The26] and is compiled with `arm-none-eabi-gcc` 10.3.1 at `-O3` without link-time optimization. A ChipWhisperer interface [OC14] controls the target. For the reference and single-write campaigns, a PicoScope records power and trigger at 500 MS/s, 270,000 samples per execution including 500 pre-trigger samples; the trigger brackets the decode loop, so a trace spans 540 µs, or approximately 3,990 clock cycles. For the masked campaigns, the trigger encloses the complete masked P2M body, which lasts approximately 350 ms; the PicoScope therefore records 500,000 samples at a measured 784 ns interval, and the analysis uses trigger-high

samples 1,000–448,058 with the preceding 1,000 samples as a control. Each campaign excludes one recorded warm-up execution, and each trace is one unaveraged execution.

The reference and single-write measurement firmware receives a serialized 608-byte ciphertext, checks and unpacks it, and computes the module products before the trigger. The trigger brackets the decode loop, after which the firmware returns the 16-byte decoded message. Re-encryption and the FO comparison of decapsulation lie outside the measured window, and assumption (A5) covers the difference in code context. The controller compares the returned message with the expected result for the capture key and query. Eligibility requires an acknowledged request, the expected message, a valid trigger swing, no digitizer overflow or clipping, and in-range power variance; this check is not available to the recovery procedure. For every dataset in Table 4, all attempted cohort keys and captures were eligible and used, with no rejected or retried cohort capture. Each source campaign additionally recorded one predeclared warm-up execution excluded from these sample counts.

The single-write firmware implements the assembly proposed by Kim and Kim [KK26]: it computes all eight decoded bits of a message byte before a single byte store. Functional comparison over 17,680 messages and 983,040 boundary pairs records no mismatch, and the compiled code contains one `strb.w` output store per byte.

Our masked implementation keeps the secret in two arithmetic shares modulo 256, refreshes key and product shares for every query, and uses a negacyclic schoolbook product. Its two-share secure AND (SecAnd) gadget follows the Ishai–Sahai–Wagner construction [ISW03], with the Boolean expression and register precharging of masked Saber [VBDK+21]. Per 32-position bitslice, two ripple additions with seven SecAnd gadgets each convert to Boolean shares before an eight-SecAnd evaluation of the predicate. There is no secret-dependent branch or address, the two Boolean output shares are combined by the controller, and a hardware random number generator supplies 408 fresh 32-bit words per query and 128 at key import. Exhaustive host tests, all 65,536 predicate inputs of the emitted ARM code, and full-path execution on 25 keys and 100 ciphertexts match the reference.

6.2 Datasets and decoder runs

Profiling maps waveform features to state scores; calibration selects Q_6 and fixes thresholds and limits; evaluation measures the fixed policies. Table 4 lists the physical datasets used in this paper. Profiling and calibration use 4,240 traces that are reused across evaluations. All profiling and calibration keys are disjoint from both evaluation cohorts. The single-write datasets repeat the key identities of the corresponding reference datasets with new executions of the protected firmware, which gives a paired implementation comparison over 10,768 traces. The masked factorial campaign interleaves 855 executions in each of four groups defined by fixed or randomized keys and ciphertexts.

Several analyses reuse the inputs of the Q_6 evaluation cohort. We distinguish four decoder runs on these inputs: the standalone policy (Table 5), which used one sequential process with the candidate and work limits of Table 3, no imposed per-key time or address-space limit, and no explicit CPU affinity; the attribution runs with full and with tied scores, the former also serving as the MITM solver run (Section 7.2); and the A* solver run (Section 7.3). These three runs use a 120 s limit per key and an 8 GiB address-space limit. These runs re-execute the decoder on identical measurements and are therefore not independent replications. The standalone decoder run took 3359.7 s in total; its median, 95th-percentile, and maximum per-key times were 2.632 s, approximately 3.2 s, and 358.5 s.

Table 4: Physical datasets. A slash separates distinct roles; a plus sign combines the listed parts. Extensions reuse the keys of their prefixes.

Dataset	Keys	Traces	Role
Profiling, natural / balanced	$112 + 112$	$784 + 784$	channel model
Profiling extension	224	672	Q_{10} features, same keys
Calibration, prefix / extension	200	$1{,}400 + 600$	select Q_6; $\tau_6, \tau_7, \tau_{10}$
Evaluation, Q_6	1,000	6,000	standalone policy
Evaluation, Q_7 / extension	1,000/7	$7{,}000 + 21$	verifier-driven policy
Repeated q_4–q_6	7	21	conditional comparison
Single-write, fixed vs. random	200	2,000	leakage test
Single-write, profiling / calibration / evaluation	224/200/1,000	1,568/1,200/6,000	protected recovery
Masked, key/ciphertext factorial	200	3,420	leakage tests
Unprotected same-rate controls	–	$6{,}000 + 2{,}000$	sensitivity controls
Masked, profiling / calibration / evaluation	224/200/60	1,568/1,200/360	masked recovery

6.3 Analysis procedures

Signal and score attribution. On the 200 calibration keys, the Q_7 exact-weight minimizer is exact for 92 keys (median blockwise distance 1), whereas permuted state identities, equal scores, and a displaced window yield no exact key (median distances 103–121).

Score intervention. The full condition keeps all 81 scores. At block i, let $\omega_i(z) = \exp(\ell_i^{(6)}(z))$. The tied condition replaces each score by

$$\widetilde{\omega}_i(z) = \frac{1}{|F_w|} \sum_{z' \in F_w} \omega_i(z'), \qquad z \in F_w, \tag{16}$$

and passes $\log \widetilde{\omega}_i$ to the unchanged decoder. The states of a two-element class then receive equal scores, each class's summed state-likelihood score is preserved, and singletons are unchanged. Each condition is calibrated on the same 200 keys after its transformation; tied log scores are rounded to 45 fractional bits (round-to-nearest, ties-to-even) before integer conversion.

Repetition comparison. The seven keys that enter the Q_{10} extension also receive one new execution of each of q_4, q_5, q_6. The repeated-observation model adds the marginal Gaussian score of these queries to the Q_7 score under conditional independence and is evaluated at the fixed thresholds τ_{10} and τ_7 without recalibration (Table 7).

Solver comparison. MITM (Algorithm 1) and an A* search receive identical scores, bounds, public inputs, and candidate verification. A* uses the exact-weight completion cost as its lower bound and applies the public relation to complete candidates. Each run is one single-threaded process on an Intel Core i7-13700F (Python 3.11.9, NumPy 1.26.4, SciPy 1.17.1, GCC 12.3.0 at `-O3`). MITM is assigned to logical CPUs 20–21, which are efficiency cores, whereas A* inherits the CPU affinity of the process that starts it.

Leakage assessment. We use samplewise two-sided Welch tests in the test vector leakage assessment (TVLA) methodology [GJJR11, SM15] with the descriptive, uncorrected threshold $|t| > 4.5$ and pre-trigger samples as a control. For single-write, 1,000 executions of one key–ciphertext pair are compared with 1,000 executions over 199 keys and the seven queries; windows guided by the disassembly keep 30 means per bit position, and the

equal-class signal-to-noise ratio (SNR) weights the 81 states equally. Gaussian classifiers are trained on one profiling cohort and tested on the other, with a 17-key label shift as the negative control. For the masked implementation, the groups FF, RF, FR, and RR fix (F) or randomize (R) the key and the ciphertext, in this order; we test FF–RF, FF–FR, FF–RR, RF–RR, and FR–RR, an alternating split of RR as an empirical null, and early–late drift controls, using pooled-mean-centered squares for second order. With 447,059 samples and independent normal statistics, approximately three null exceedances are expected, and the smallest standardized mean difference that reaches $|t| = 4.5$ with 855 traces per group is 0.22. The masked attack model uses centered first-order values and products from store or multiply-accumulate regions, is selected by the validation negative log-likelihood (NLL) of the true state on $56 + 56$ balanced profiling keys, and is refitted on both cohorts. Both protected evaluations use the Q_6 limits with 120 s and 8 GiB per key, and ground truth is unblinded only after all decisions terminate.

6.4 Metrics

The statistical unit for recovery is an evaluation key. For key k with status ρ_k, released secret $\widehat{s}_k$, and true secret $s_k^\star$, we report

$$R_k = \mathbf{1}[\rho_k = \textsc{Certified} \wedge \widehat{s}_k = s_k^\star], \qquad F_k = \mathbf{1}[\rho_k = \textsc{Certified} \wedge \widehat{s}_k \neq s_k^\star], \tag{17}$$

where equality is byte-for-byte equality of the serialized secret; abstentions count in neither event. For $y = \sum_k R_k$ among N keys, we report the exact one-sided 95% lower Clopper–Pearson bound [CP34] $\mathrm{Beta}^{-1}(0.05; y, N - y + 1)$, and the two-sided exact interval for the masked attack. Top-one accuracy is the fraction of correctly assigned blocks, and macro accuracy is the unweighted mean recall over the 81 states. Search times exclude feature extraction and Gaussian scoring, which we report separately, and the 95th percentile of ordered values $W_{(1)} \leq \cdots \leq W_{(N)}$ is $W_{(\lceil 0.95(N-1) \rceil + 1)}$.

7 Results

7.1 Complete PKE-secret recovery

Table 5 summarizes both policies. The standalone Q_6 policy recovers 999 of 1,000 secrets exactly; the remaining key exhausts its work bound and yields no key. The verifier-driven policy recovers all 1,000 secrets of its cohort: 993 are certified after seven traces and seven after the three-trace extension. Neither policy releases an incorrect key. The verifier-driven policy spends its additional traces on 0.7% of the keys and reduces the median and 95th-percentile decoder work. Its first stage certifies fewer keys than the Q_6 policy despite one more trace per key. The two policies use different cohorts, and all seven Q_7 abstentions completed the calibrated τ_7 region without reaching the work bound; their true secrets lay outside that region. Since every certified key lies in its region, at most one Q_6 key and at most seven Q_7 keys fall outside their regions. By Proposition 10, the miscoverage probability of the selected Q_6 configuration is below 0.48% and that of the Q_7 stage below 1.32%, each with 95% confidence; the first value is comparable to the a priori bound $1/201 \approx 0.50\%$ of Proposition 8 for a fixed configuration.

7.2 What waveform scores add beyond binary transcripts

Figure 1 compares the full and the tied scores on the same 1,000 measured Q_6 inputs under common 120 s and 8 GiB limits. With full scores, the local maxima already form the exact secret for 29 keys, the exact-weight minimizer is exact for 113 keys, and certified enumeration recovers 998 keys, with two time-limit expirations. With tied scores, these

Table 5: Evaluation of the two fixed acquisition and recovery policies on disjoint cohorts. Decoder work counts enumeration and verification in every entered stage.

Metric	Standalone Q_6	Verifier-driven Q_7–Q_{10}
Evaluation keys	1,000	1,000
Certification stage	999 at Q_6	993 at Q_7, 7 at Q_{10}
Exact complete-secret recovery	999/1,000	1,000/1,000
False acceptances	0	0
Mean / maximum traces per key	6/6	7.021/10
Decoder work, median / 95th percentile	3,759.5/885,333	633.5/30,565
One-sided exact 95% lower bound	0.9952650	0.9970088

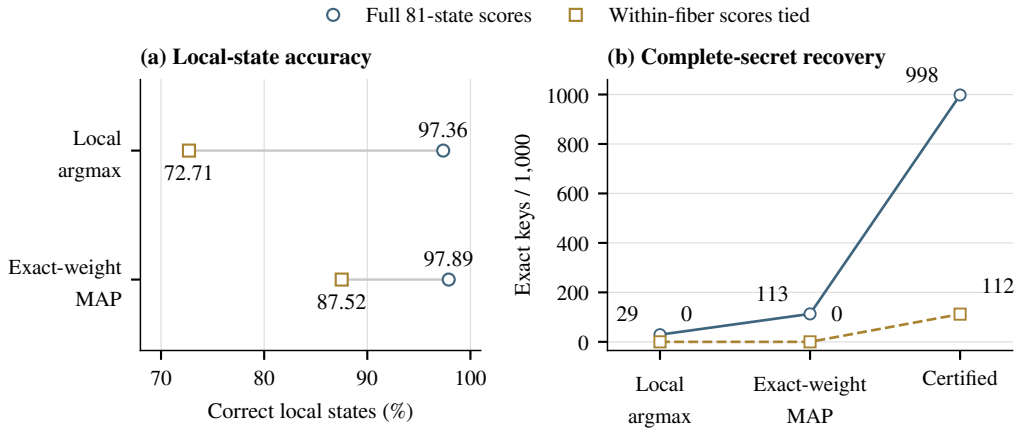

Figure 1: Score intervention on common measured Q_6 inputs. (a) Local-state accuracy over 128 blocks per key. (b) Exact complete-secret recovery at each decoding stage. Tying preserves each class's summed state-likelihood score and removes the ordering of its states.

counts are 0, 0, and 112, with 832 timeouts, 55 address-space-limit terminations, and one work-bound termination. Neither condition releases an incorrect key. The local accuracy rises from 97.36% to 97.89% under the weights with full scores, and from 72.71% to 87.52% with tied scores. For the 112 keys certified with tied scores, the interval from the first verified candidate to the completed certification has median 0.0224 s and maximum 0.0301 s.

Proposition 3 provides theoretical insight into this performance gap. With tied scores and correctly identified classes, all secrets consistent with the class sequence and the weights have equal cost, and the geometric mean of this set's size is $2^{27.2}$. Certification must exclude every other member of the set; Algorithm 1 does so through the residual index and full checks rather than through cost ordering. By Corollary 1, the public relation suffices to identify the secret. The intervention therefore shows that, for this decoder and these limits, removing the within-class scores reduces certified recovery from 998 to 112 keys; it does not show that such information is necessary for every transcript-based attack. Nor does it localize the physical origin of the within-class distinctions, which we describe only as information not captured by the local binary transcripts. A paired-key measurement that flips one sign within a weight-neutral class and fixes all other coefficients could help to localize it.

Table 6: Complete-secret solver cost on the Q_6 evaluation inputs with one process, one numerical thread, 120 s, and 8 GiB per key. The certification delay is the time from the first verified candidate to the completed minimum-cost decision. Triples give the median, 95th percentile, and maximum.

Metric	MITM	A*
Exact recovery	998/1,000	930/1,000
First verified candidate (s)	4.021/4.921/64.853	2.969/33.965/119.702
Certified minimum (s)	4.053/4.955/64.884	2.981/34.021/119.824
Certification delay (s)	0.029/0.035/0.050	0.012/0.040/0.205
Final decision (s)	4.053/5.037/120.916	3.085/120.082/120.294
Final decision, total (s)	4,702.772	15,488.546
Peak resident memory (GiB)	2.159	1.099
Timeouts / false acceptances	2/0	70/0

Table 7: Conditional recovery of the seven keys not certified under Q_7. The extension uses the fitted Q_{10} model.

Strategy	Added traces per key	Threshold	Work bound	Recovery
Additional computation	0	$2\tau_7$	5×10^8	7/7
Repeat q_4–q_6	3	τ_{10}	5×10^7	2/7
Repeat q_4–q_6	3	τ_7	5×10^7	5/7
Fixed q_8–q_{10} extension	3	τ_{10}	5×10^7	7/7

7.3 Solver and certification cost

Table 6 reports the solver comparison on the same inputs. MITM recovers 998 keys and A* recovers 930, both without false acceptance. A* has the lower median time and approximately half the peak memory, whereas MITM has a shorter tail, with 2 time-limit expirations compared to 70 for A*. In these runs, MITM recovers 68 more secrets at the cost of 1.060 GiB of additional peak memory. The two runs used different CPU assignments, so these timings characterize the two configurations as executed and are not a controlled comparison of the algorithms. Completing the minimum-cost decision after the first verified candidate adds a median of 0.029 s for MITM and 0.012 s for A*. For MITM, the exact-weight dynamic program and the completion tables take medians of 0.405 s and 3.554 s per key and dominate the search; enumeration with the public relation takes a median of 0.017 s (maximum 60.877 s), and KEM verification is negligible. Feature extraction and Gaussian scoring take 11.580 s in total for all 1,000 keys.

7.4 Fixed extension, repeated observations, and computation

Table 7 compares strategies for the seven keys not certified under Q_7. The fixed extension recovers all seven keys, and so does additional computation on the original seven traces. The seven Q_7 abstentions therefore reflect the calibrated threshold rather than missing information in the seven traces: widening the threshold to $2\tau_7$ recovers all seven from the original traces. The repeated-observation model recovers two or five keys, depending on the threshold; because this model is not calibrated separately, the difference from the extension cannot be attributed to query geometry alone. With seven keys, these comparisons are descriptive. No strategy releases an incorrect key.

7.5 Countermeasures under a six-execution budget

Single-write assembly. The fixed-versus-random TVLA reaches $\max|t| = 30.35$, with 19,102 of 270,000 samples above 4.5; the pre-trigger control reaches $\max|t| = 2.41$. The

peak equal-class SNR is 0.01019, and held-out top-one accuracy is 11.24% and 3.75% in the two directions (0.47% and 1.09% with shifted labels). On the evaluation cohort, the local maxima match 1,632 of 128,000 states (1.28%, close to $1/81 \approx 1.23\%$), the exact-weight minimizer matches 18,176, and certified search recovers no secret (999 timeouts, one address-space-limit termination) and releases no incorrect key. Single-write assembly changes the loop structure in addition to the store, so the loss of accuracy cannot be attributed to the store alone. Averaging k executions of one chosen ciphertext multiplies the SNR by k under independent noise and a stable signal [MOP07]; as an arithmetic estimate, not a prediction of attack success, a per-feature SNR near one would require on the order of 100 executions per query. Thus the evaluated attack recovered no complete secret under the six-execution budget and stated limits.

Two-share masked P2M. Across 3,420 executions, the first-order exceedance counts of the five data contrasts are 0, 0, 0, 0, and 3 (FR–RR), against 2 for the empirical null and 68 for the drift control. For centered squares, the counts are 0, 0, 2, 0, and 3, against 1 and 6,320; all pre-trigger counts are zero. At the same 784 ns sampling interval, a natural unprotected control with 3,000 traces per group reaches $\max |t| = 6.0783$ with 55 crossings, and a high-contrast control with 1,000 traces per group reaches 5.4040 with 13 crossings; both have zero pre-trigger crossings. Neither meets the predeclared sensitivity criterion of peak $|t| \geq 8$, at least 10 crossings, and zero pre-trigger crossings. The six-query attack recovers 0 of 60 secrets without false acceptance (two-sided exact 95% interval $[0, 5.963\%]$), and all searches reach the time limit. The selected model (48 store-region features per query) has a held-out true-state NLL of 4.9976 nats, worse than the uniform predictor's $\ln 81 \approx 4.3944$ nats. Its top-one and macro accuracies are 0.558% and 1.300%; local and exact-weight predictions match 23 and 1,160 of 7,680 states. The model did not enable recovery within this budget, which does not establish that the traces contain no exploitable information. Emulating one valid input of the P2M body with the Unicorn CPU emulator counts 1,425,806 ARM instructions for the masked implementation and 270,459 for the reference (5.272×), excluding protocol, communication, trigger, and random-word generation; flash load size and allocated RAM increase by 17.8% and 40.7%, respectively.

8 Discussion

Information and search. Noiseless six-query transcripts, the weights, and the public key determine the secret (Corollary 1); the $2^{27.2}$ consistent secrets quantify the residual ambiguity, not the search cost. For our decoder and limits, the within-class scores raise certified recovery from 112 to 998 keys, and Theorem 2 bounds false certification without the channel model. The public relation also resolves larger residual sets in principle: the subschedules $(q_1, q_4, q_5, q_6, q_7)$ and (q_1, q_4, q_6, q_7) of Q_7 leave 51.1 and 127.2 bits on average over the same 2,000 keys. Whether measured scores support certified recovery with fewer than six traces remains an open question.

Countermeasures. Kim and Kim propose single-write assembly, randomized bit order, masking, and temporal hiding [KK26]; related defenses include masking-friendly designs [KKV24], execution-order shuffling [XWT25], and message randomization through ciphertext malleability [Ber25]. Against single-write assembly, the evaluated attack recovered no secret within six executions, although measurable dependence remains that repeated queries can accumulate.

Limitations. The physical evaluation uses one STM32F415 at one clock frequency. The measurement firmware does not execute the complete decapsulation path, so the success rates on the KEM rest on (A5); verifying it requires measurements of decapsulation with

the encoded ciphertexts. Proposition 7 relies on (A3), Theorem 1 on the uniform-matrix model (A2), and Proposition 8 on (A4); because the calibration keys also selected Q_6, only Propositions 9 and 10 apply to τ_6. Proposition 6 concerns the displayed prefix and the distance-two objective, and the conditional comparison concerns seven keys. The solver runs used different CPU assignments. The countermeasure results hold for the stated six-execution budget, features, and limits. The same-rate unprotected controls did not meet the predeclared sensitivity criterion, so the masked results report the observed pointwise statistics and bounded recovery rate; they do not establish leakage absence or non-detection at a validated sensitivity. Repeated-query and cross-time higher-order evaluation, explainable localization [GO24], and a probing-model or information-theoretic analysis [ISW03, SMY09] remain future work.

9 Conclusion

Poly2MsgLeaks recovers the complete 512-coefficient PKE secret of SMAUG-T from six physical TiMER P2M traces for 999 of 1,000 keys, and its verifier-driven policy recovers all 1,000 keys of a second cohort at a mean of 7.021 traces, both without false acceptance. Noiseless six-query transcripts and the exact weights leave 27.2 bits of ambiguity on average; for our decoder and limits, the within-class scores raise certified recovery from 112 to 998 keys. Certification bounds false release independently of the channel model, and calibration bounds the miscoverage probability for exchangeable keys. Against single-write assembly and our two-share masked P2M, the evaluated attack recovered no complete secret within the tested six-execution budget.

A Proof of Lemma 5

Proof of Lemma 5. Let $u = x + 1 \in R$, where $R = \mathbb{Z}_{1024}[x]/(x^{256} + 1)$ and $|R| = 2^{2560}$.

Step 1: units. Evaluation at $x = -1$ induces $R/(u) \cong \mathbb{Z}_{1024}/((-1)^{256}+1) = \mathbb{Z}_{1024}/(2) \cong \mathbb{F}_2$, so (u) is a maximal ideal that contains 2. In $\mathbb{Z}[x]$, $(x+1)^{256} = x^{256}+1+\sum_{k=1}^{255}\binom{256}{k}x^k$, and every $\binom{256}{k}$ with $1 \le k \le 255$ is even. Hence $u^{256} = 2g$ in R with $g = \sum_{k=1}^{255}\frac{1}{2}\binom{256}{k}x^k$ and $g(-1) = \frac{1}{2}\big((1-1)^{256} - 2\big) = -1$. Consequently $u^{2560} = 2^{10}g^{10} = 0$, so u is nilpotent and every element of (u) is nilpotent. An element $f \notin (u)$ is a unit, because $f = f(-1) - m$ with $f(-1)$ odd and $m \in (u)$ nilpotent, and an odd residue is a unit modulo 1024. In particular, g is a unit and $(2) = (u^{256})$.

Step 2: chain structure. Every nonzero $a \in R$ has a valuation $v(a) = \max\{i : a \in (u^i)\} < 2560$, and $a = u^{v(a)}w$ with $w \notin (u)$, so w is a unit. Hence every ideal $I \neq 0$ equals (u^t) with $t = \min_{a\in I} v(a)$. For i below the nilpotency index of u, multiplication by u^i induces a surjection $R/(u) \to (u^i)/(u^{i+1})$, and the quotient is nonzero: otherwise $u^i = u^{i+1}r$, so $u^i(1-ur) = 0$ and $u^i = 0$ because $1 - ur$ is a unit. Each quotient therefore has two elements, and since $|R| = 2^{2560}$, the nilpotency index is 2560 and $|(u^i)| = 2^{2560-i}$.

Step 3: valuations of secret differences. Reduction modulo 2 maps R onto $R/(2) = R/(u^{256}) \cong \mathbb{F}_2[x]/((x+1)^{256})$. If a nonzero component $a \in \{d_0, d_1\}$ has an odd coefficient, its reduction is nonzero, so $a \notin (u^{256})$ and $v(a) \le 255$. Otherwise all coefficients of a lie in $\{-2, 0, 2\}$, so $a = 2c$ with $c \neq 0$ having coefficients in $\{-1, 0, 1\}$; then c has an odd coefficient, $v(c) \le 255$, and $v(a) = v(u^{256}g^{-1}c) = 256 + v(c) \le 511$.

Step 4: the image. The image of ϕ_d is the ideal $d_0R + d_1R$, which by Step 2 equals (u^t) with $t = \min\{v(d_j) : d_j \neq 0\} \le 511$. Its size is $2^{2560-t} \ge 2^{2049}$. □

B Exhaustive Computations

The combinatorial counts use exact integers; logarithms and summary statistics use binary64 arithmetic. The supplementary material provides the scripts.

Predicate family (Fact 1). Evaluating each descriptor with (6) and (5) yields the same 81-bit table; deduplication leaves 60,898 tables, including two constants.

Transcript classes (Proposition 2). The 81×7 matrix of Q_7 has 182 nearest pairs. Deleting q_2 yields 33 singleton and 24 two-element classes, four of them weight-neutral; Table 2 summarizes all 3,240 state pairs.

Minimal repair (Proposition 6). No pair among the 47,684 separation sets covers all 182 nearest pairs.

References

[AB23] Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

[ACLZ20] Dorian Amiet, Andreas Curiger, Lukas Leuenberger, and Paul Zbinden. Defeating NewHope with a single trace. In *Post-Quantum Cryptography – PQCrypto 2020*, volume 12100 of *Lecture Notes in Computer Science*, pages 189–205. Springer, 2020.

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange – A New Hope. In *25th USENIX Security Symposium*, pages 327–343. USENIX Association, 2016.

[BCVCV26] Alexandre Berzati, Andersson Calle Viera, Maya Chartouny, and David Vigilant. Simple power analysis assisted chosen cipher-text attack on ML-KEM. In *Constructive Approaches for Security Analysis and Design of Embedded Systems – CASCADE 2025*, volume 15952 of *Lecture Notes in Computer Science*, pages 3–26. Springer, 2026.

[Ber25] Pierre-Augustin Berthet. Ciphertext malleability in lattice-based KEMs as a countermeasure to side channel analysis. *Fundamenta Informaticae*, 194(4):1–28, 2025. Article 1.

[BK12] Andrey Bogdanov and Ilya Kizhvatov. Beyond the limits of DPA: Combined side-channel collision attacks. *IEEE Transactions on Computers*, 61(8):1153–1164, 2012.

[CCC+24a] Jung Hee Cheon, Hyeongmin Choe, Joongeun Choi, Dongyeon Hong, Jeongdae Hong, Chi-Gon Jung, Honggoo Kang, Janghyun Lee, Seonghyuck Lim, Aesun Park, Seunghwan Park, Jungjoo Seo, Hyoeun Seong, and Junbum Shin. SMAUG-T: The key exchange algorithm based on module-LWE and module-LWR. Specification, version 4.0, https://hmchoe0528.github.io/assets/manuscripts/SMAUG_T_spec_Oct24_v4.0.pdf, October 2024. KpqC Round 2 submission document.

[CCC+24b] Jung Hee Cheon, Hyeongmin Choe, Joongeun Choi, Dongyeon Hong, Jeongdae Hong, Chi-Gon Jung, Honggoo Kang, Janghyun Lee, Seonghyuck Lim, Aesun Park, Seunghwan Park, Hyoeun Seong, and Junbum Shin. SMAUG-T: The key exchange algorithm based on module-LWE and module-LWR. Specification, version 3.0, https://kpqc.or.kr/images/pdf/SMAUG-T_Document.pdf, February 2024. KpqC Round 2 submission document, February 23, 2024.

[CCHY24] Jung Hee Cheon, Hyeongmin Choe, Dongyeon Hong, and MinJune Yi. SMAUG: Pushing lattice-based key encapsulation mechanisms to the limits. In *Selected Areas in Cryptography – SAC 2023*, volume 14201 of *Lecture Notes in Computer Science*, pages 127–146. Springer, 2024.

[CCSS24] Jung Hee Cheon, Hyeongmin Choe, Jungjoo Seo, and Hyoeun Seong. SMAUG(-T), revisited: Timing-secure, more compact, less failure. *IEEE Access*, 12:188386–188397, 2024.

[Cha85] David Chase. Code combining – a maximum-likelihood decoding approach for combining an arbitrary number of noisy packets. *IEEE Transactions on Communications*, 33(5):385–393, 1985.

[CP34] Charles J. Clopper and Egon S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

[CRR03] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In *Cryptographic Hardware and Embedded Systems – CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2003.

[CT01] Giuseppe Caire and Daniela Tuninetti. The throughput of hybrid-ARQ protocols for the Gaussian collision channel. *IEEE Transactions on Information Theory*, 47(5):1971–1988, 2001.

[Din05] Hai Q. Dinh. Negacyclic codes of length 2^s over Galois rings. *IEEE Transactions on Information Theory*, 51(12):4252–4262, 2005.

[DLP04] Hai Q. Dinh and Sergio R. López-Permouth. Cyclic and negacyclic codes over finite chain rings. *IEEE Transactions on Information Theory*, 50(8):1728–1744, 2004.

[DSDGR20] Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In *Advances in Cryptology – CRYPTO 2020*, volume 12171 of *Lecture Notes in Computer Science*, pages 329–358. Springer, 2020.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554. Springer, 1999.

[GJJR11] Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi. A testing methodology for side-channel resistance validation. NIST Non-Invasive Attack Testing Workshop, https://csrc.nist.gov/csrc/media/events/non-invasive-attack-testing-workshop/documents/08_goodwill.pdf, 2011.

[GM23] Qian Guo and Erik Mårtensson. Do not bound to a single position: Near-optimal multi-positional mismatch attacks against Kyber and Saber. In *Post-Quantum Cryptography – PQCrypto 2023*, volume 14154 of *Lecture Notes in Computer Science*, pages 291–320. Springer, 2023.

[GNJ26] Qian Guo, Denis Nabokov, and Thomas Johansson. Unlocking the true potential of decryption failure oracles: A hybrid adaptive-LDPC attack on ML-KEM using imperfect oracles. In *35th USENIX Security Symposium*, pages 2405–2423. USENIX Association, 2026.

[GNNJ23] Qian Guo, Denis Nabokov, Alexander Nilsson, and Thomas Johansson. SCA-LDPC: A code-based framework for key-recovery side-channel attacks on post-quantum encryption schemes. In *Advances in Cryptology – ASIACRYPT 2023*, volume 14441 of *Lecture Notes in Computer Science*, pages 203–236. Springer, 2023.

[GO24] Si Gao and Elisabeth Oswald. A novel framework for explainable leakage assessment. In *Advances in Cryptology – EUROCRYPT 2024*, volume 14653 of *Lecture Notes in Computer Science*, pages 221–250. Springer, 2024.

[Hag88] Joachim Hagenauer. Rate-compatible punctured convolutional codes (RCPC codes) and their applications. *IEEE Transactions on Communications*, 36(4):389–400, 1988.

[HG07] Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 150–169. Springer, 2007.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki–Okamoto transformation. In *Theory of Cryptography – TCC 2017*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371. Springer, 2017.

[HHP+21] Mike Hamburg, Julius Hermelink, Robert Primas, Simona Samardjiska, Thomas Schamberger, Silvan Streit, Emanuele Strieder, and Christine van Vredendaal. Chosen ciphertext k-trace attacks on masked CCA2 secure Kyber. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(4):88–113, 2021.

[HSMP25] Julius Hermelink, Silvan Streit, Erik Mårtensson, and Richard Petri. A generic framework for side-channel attacks against LWE-based cryptosystems. In *Advances in Cryptology – EUROCRYPT 2025*, volume 15608 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2025.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.

[KK26] Gwang-Sik Kim and Young-Sik Kim. Side-channel leakage assessment of SMAUG-T: Exploiting Hamming weight patterns in polynomial-to-message conversion. In *Information Security and Cryptology – ICISC 2025*, volume 16487 of *Lecture Notes in Computer Science*, pages 198–219. Springer, 2026.

[KKV24] Suparna Kundu, Angshuman Karmakar, and Ingrid Verbauwhede. On the masking-friendly designs for post-quantum cryptography. In *Security, Privacy, and Applied Cryptography Engineering – SPACE 2023*, volume 14412 of *Lecture Notes in Computer Science*, pages 162–184. Springer, 2024.

[LCS+25] Jinnuo Li, Chi Cheng, Muyan Shen, Peng Chen, Qian Guo, Dongsheng Liu, Liji Wu, and Jian Weng. Grafted trees bear better fruit: An improved multiple-valued plaintext-checking side-channel attack against Kyber. In *2025 Design, Automation & Test in Europe Conference (DATE)*, pages 1–7. IEEE, 2025.

[May21] Alexander May. How to meet ternary LWE keys. In *Advances in Cryptology – CRYPTO 2021*, volume 12826 of *Lecture Notes in Computer Science*, pages 701–731. Springer, 2021.

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, 2007.

[MRD+26] Yashaswini Makaram, Davis Ranney, Adam A. Ding, David Kaeli, and Yunsi Fei. Formal methods-assisted chosen ciphertext attacks on PQC CRYSTALS-Kyber using electromagnetic emanations. In *2026 Design, Automation & Test in Europe Conference (DATE)*, pages 1–7. IEEE, 2026.

[NDGJ21] Kalle Ngo, Elena Dubrova, Qian Guo, and Thomas Johansson. A side-channel attack on a masked IND-CCA secure Saber KEM implementation. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(4):676–707, 2021.

[OC14] Colin O'Flynn and Zhizhang Chen. ChipWhisperer: An open-source platform for hardware embedded security research. In *Constructive Side-Channel Analysis and Secure Design – COSADE 2014*, volume 8622 of *Lecture Notes in Computer Science*, pages 243–260. Springer, 2014.

[PP19] Peter Pessl and Robert Primas. More practical single-trace attacks on the number theoretic transform. In *Progress in Cryptology – LATINCRYPT 2019*, volume 11774 of *Lecture Notes in Computer Science*, pages 130–149. Springer, 2019.

[PPM17] Robert Primas, Peter Pessl, and Stefan Mangard. Single-trace side-channel attacks on masked lattice-based encryption. In *Cryptographic Hardware and Embedded Systems – CHES 2017*, volume 10529 of *Lecture Notes in Computer Science*, pages 513–533. Springer, 2017.

[PRI26] Paco Poilbout, Thomas Roche, and Laurent Imbert. Handling noisy plaintext checking oracles with SPiRiT: Application to Kyber. In *Post-Quantum Cryptography – PQCrypto 2026*, volume 16492 of *Lecture Notes in Computer Science*, pages 311–341. Springer, 2026.

[QCZ+21] 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. In *Advances in Cryptology – ASIACRYPT 2021*, volume 13093 of *Lecture Notes in Computer Science*, pages 92–121. Springer, 2021.

[QGZ+26] Yuhan Qian, Jing Gao, Yuchen Zhong, Yaoling Ding, Jingjie Wu, Weiping Gong, Zihe Lin, and An Wang. One time is enough: Chosen-ciphertext side-channel attack on ML-KEM cryptosystems. In *Applied Cryptography and Network Security Workshops – ACNS 2025*, volume 15654 of *Lecture Notes in Computer Science*, pages 23–40. Springer, 2026.

[QWC+24] Kexin Qiao, Zhaoyang Wang, Heng Chang, Siwei Sun, Zehan Wu, Junjie Cheng, Changhai Ou, An Wang, and Liehuang Zhu. A closer look at the belief propagation algorithm in side-channel attack on CCA-secure PQC KEM. *Science China Information Sciences*, 67(11):212302, 2024.

[RBSRC22] Prasanna Ravi, Shivam Bhasin, Sujoy Sinha Roy, and Anupam Chattopadhyay. On exploiting message leakage in (few) NIST PQC candidates for practical message recovery attacks. *IEEE Transactions on Information Forensics and Security*, 17:684–699, 2022.

[RRD+23] Gokulnath Rajendran, Prasanna Ravi, Jan-Pieter D'Anvers, Shivam Bhasin, and Anupam Chattopadhyay. Pushing the limits of generic side-channel attacks on LWE-based KEMs – parallel PC oracle attacks on Kyber KEM and beyond. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(2):418–446, 2023.

[RSK+26] Madhumitha Ramaswamy, Rishav Saha, Suparna Kundu, Anupam Golder, Angshuman Karmakar, and Debayan Das. POSTER: SMAUG-SCA: Machine learning based power side-channel attack on SMAUG-T. In *Proceedings of the ACM Asia Conference on Computer and Communications Security*, pages 1895–1897. ACM, 2026.

[RSRCB20] Prasanna Ravi, Sujoy Sinha Roy, Anupam Chattopadhyay, and Shivam Bhasin. Generic side-channel attacks on CCA-secure lattice-based PKE and KEMs. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(3):307–335, 2020.

[SCZ+22] Muyan Shen, Chi Cheng, Xiaohan Zhang, Qian Guo, and Tao Jiang. Find the bad apples: An efficient method for perfect key recovery under imperfect SCA oracles – a case study of Kyber. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(1):89–112, 2022.

[SKL+20] Bo-Yeon Sim, Jihoon Kwon, Joohee Lee, Il-Ju Kim, Tae-Ho Lee, Jaeseung Han, Hyojin Yoon, Jihoon Cho, and Dong-Guk Han. Single-trace attacks on message encoding in lattice-based KEMs. *IEEE Access*, 8:183175–183191, 2020.

[SLZ24] Mingyao Shao, Yuejun Liu, and Yongbin Zhou. Pairwise and parallel: Enhancing the key mismatch attacks on Kyber and beyond. In *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, pages 548–559. ACM, 2024.

[SLZ+27] 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. In *Computer Security – ESORICS 2026*, volume 16969 of *Lecture Notes in Computer Science*, pages 341–362. Springer, 2027.

[SM15] Tobias Schneider and Amir Moradi. Leakage assessment methodology—a clear roadmap for side-channel evaluations. In *Cryptographic Hardware and Embedded Systems – CHES 2015*, volume 9293 of *Lecture Notes in Computer Science*, pages 495–513. Springer, 2015.

[SMY09] François-Xavier Standaert, Tal G. Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.

[SPH22] Bo-Yeon Sim, Aesun Park, and Dong-Guk Han. Chosen-ciphertext clustering attack on CRYSTALS-KYBER using the side-channel leakage of Barrett reduction. *IEEE Internet of Things Journal*, 9(21):21382–21397, 2022.

[The26] The SMAUG-T Team. SMAUG-T: Reference implementation and specification. Official distribution page, https://kpqc.cryptolab.co.kr/smaug-t; specification 260521 and reference implementation version 1.2.0, 2026. TiMER parameter set.

[TUX+23] Yutaro Tanaka, Rei Ueno, Keita Xagawa, Akira Ito, Junko Takahashi, and Naofumi Homma. Multiple-valued plaintext-checking side-channel attacks on post-quantum KEMs. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(3):473–503, 2023.

[UXT+21] Rei Ueno, Keita Xagawa, Yutaro Tanaka, Akira Ito, Junko Takahashi, and Naofumi Homma. Curse of re-encryption: A generic power/EM analysis on post-quantum KEMs. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(1):296–322, 2021.

[VBDK+21] Michiel Van Beirendonck, Jan-Pieter D'Anvers, Angshuman Karmakar, Josep Balasch, and Ingrid Verbauwhede. A side-channel-resistant implementation of SABER. *ACM Journal on Emerging Technologies in Computing Systems*, 17(2):1–26, April 2021. Article 10.

[VCGRS13] Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renauld, and François-Xavier Standaert. An optimal key enumeration algorithm and its application to side-channel attacks. In *Selected Areas in Cryptography – SAC 2012*, volume 7707 of *Lecture Notes in Computer Science*, pages 390–406. Springer, 2013.

[VCGS13] Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Security evaluations beyond computing power: How to analyze side-channel attacks you cannot mount? In *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 126–141. Springer, 2013.

[VCGS14] Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Soft analytical side-channel attacks. In *Advances in Cryptology – ASIACRYPT 2014*, volume 8873 of *Lecture Notes in Computer Science*, pages 282–296. Springer, 2014.

[Wun19] Thomas Wunderer. A detailed analysis of the hybrid lattice-reduction and meet-in-the-middle attack. *Journal of Mathematical Cryptology*, 13(1):1–26, 2019.

[XPSR+22] Zhuang Xu, Owen Pemberton, Sujoy Sinha Roy, David Oswald, Wang Yao, and Zhiming Zheng. Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of Kyber. *IEEE Transactions on Computers*, 71(9):2163–2176, 2022.

[XWT25] Dejun Xu, Kai Wang, and Jing Tian. A hardware-friendly shuffling countermeasure against side-channel attacks for Kyber. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 72(3):504–508, 2025.

[YHH25] Sung-Hwan Yoo, Jaeseung Han, and Dong-Guk Han. Simple power analysis and countermeasure for SMAUG-T multiplication operation. *Journal of the Korea Institute of Information Security & Cryptology*, 35(1):25–35, 2025. In Korean.