Packing Stern into L2:
GPU-Accelerated Primal-Shortened ISD for High-Codimension Binary Goppa Key Recovery

Raullen Chai Xinxin Fan
MachineFi Labs
{raullen, xinxin}@machinefi.com

November 23, 2026

Abstract

Primal-Shortened Information-Set Decoding (ISD) with quadratic polar lifting recently pushed TII McEliece Track-2 key recovery to $mt = 126$ (`TII-228`, 2,463 CPU core-hours), with over 99% of runtime spent harvesting weight-$(2t + 1)$ codewords. Porting MiniMax's Mode-6 Stern sampler directly to one block per SM leaves 1.0–1.5 GB of hash tables active and stalls on global memory. We pack 128 syndrome bits into one 16-byte `ulonglong2` without left-triple indices (lossless for $mt \leq 145$, and extended to $mt \leq 192$ via truncated range pre-filtering), share $G \in \{4, 8\}$ cooperative tables inside GPU L2 cache, batch $64G$ random systematic reductions per launch, and widen the shortening prefix n' to cut modeled GPU-hours by 34%–47%.

Across five Track-2 challenges ($mt = 108 \rightarrow 153$), every recovered private key passes both the official SageMath checker and a standalone pure-Python verifier. Eight Modal L40S GPUs solve `TII-228` ($mt = 126$) in 18.9 min (2.15 GPU-hours, 3.0× faster in wall-clock than MiniMax's EPYC cluster). We further recover keys for the previously undocumented challenges `TII-237` ($mt = 135$, 22.1 L4 GPU-hours), `TII-235` ($mt = 144$, 39.62 L40S GPU-hours, 17.97 s lift), and `TII-185` ($mt = 153$, 140.4 L40S + $\approx$ 60 L4 GPU-hours, 39.01 s lift). These results do not demonstrate practical recovery against Classic McEliece parameters ($mt \geq 768$).

Keywords: McEliece cryptosystem, binary Goppa codes, information-set decoding, GPU cryptanalysis, TII McEliece Challenges.

1 Introduction

The McEliece cryptosystem [1] conceals a binary Goppa code $\Gamma(L, g)$ behind a systematic parity-check matrix $H_{\text{pub}} \in \mathbb{F}_2^{mt \times n}$. Although Classic McEliece reached Round 4 of the NIST post-quantum standardization process before NIST selected HQC in March 2025, concrete structural key-recovery margins on smaller parameters remained sparsely mapped until the Technology Innovation Institute (TII) published 204 non-toy Track-2 public-key recovery challenges alongside two toy verification instances [2].

In November 2026, MiniMax [3] extended Track-2 recoveries to 94 keys up to codimension $mt = 126$ using Primal-Shortened ISD (Route A) alongside an auxiliary HOVER pipeline (Route B). What makes Route A structurally unusual is its runtime asymmetry. On `TII-245` ($mt = 108$), SageMath recovers (g, L) from harvested codewords in under ten seconds, whereas an OpenMP C sampler (`minwords_mt.c`) consumes over 99% of wall-clock time searching for weight-$(2t + 1)$ vectors in a shortened parity-check kernel. The algebra is already cheap. Everything that limits codimension scaling sits inside a memory-bound collision search.

We leave MiniMax's algebraic recovery algorithm and Mode-6 Stern parameters ($p = 3+3, l = 17$, 16 KB occupancy bitmap) intact, and redesign the harvesting stage around two coupled

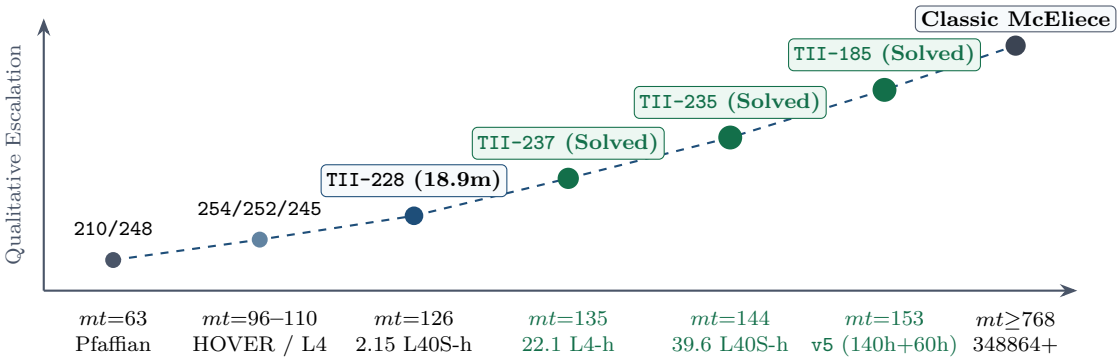


Figure 1: Qualitative escalation of publicly documented TII Track-2 key recoveries by codimension mt. Our five verified recoveries (`TII-245`, `TII-228`, `TII-237`, `TII-235`, `TII-185`) span $mt = 108 \to 153$, with `TII-185` extending past the 128-bit lossless packing ceiling via truncated 16-byte range pre-filtering (`v5`).

constraints: GPU L2 cache capacity and the shortening-window trade-off. Sizing $G = 4$ or $G = 8$ cooperative hash tables below nominal L2 capacity reduces global-memory pressure; widening the shortened prefix n' beyond narrow CPU-oriented choices trades a modest increase in $\mathbb{F}_2$ matrix dimension for a 10–25× jump in codeword supply, yielding a modeled 34%–47% reduction in GPU-hours. Together, these changes compress `TII-228` ($mt = 126$) from 57.1 min on 2,048–5,120 EPYC cores to 18.9 min on eight L40S GPUs, and push verified Track-2 key recovery through `TII-237` ($mt = 135$) and `TII-235` ($mt = 144$, punctured at 10 positions) to `TII-185` ($mt = 153$).

Because codimension mt does not order difficulty monotonically across arbitrary (n, m, t, n') tuples—MiniMax's `TII-250` ($mt = 120, k' = 347$) required far more search work than `TII-237` ($mt = 135, k' = 243$)—we treat $mt = 153$ as a concrete parameter milestone rather than a universal difficulty claim. In packing terms, `TII-235` ($mt = 144$) occupies 127 of 128 bits in our lossless 16-byte `ulonglong2` format (`v4`, ceiling $mt = 145$), while `TII-185` ($mt = 153$) crosses that ceiling by 8 bits and validates our truncated 16-byte range pre-filter (`v5`) without inflating the L2-sized hash table.

2 Prior Work and the Track-2 Landscape

During 2026, two distinct families of structural attacks drove Track-2 progress: pure algebraic vanishing methods and hybrid combinatorial-algebraic harvesting.

Bardet, Lemoine, and Tillich [4] broke instances up to `TII-210` ($mt = 63$) via Pfaffian relations, and Hemmert [5] reached `TII-248` ($mt = 63$) using higher-order vanishing (HOV). Following hold-out improvements by Vedenev [6] and Weis [7] on `TII-253` ($n = 214, m = 8, t = 9, mt = 72$), Saarinen introduced Frobenius-reduced HOVER to solve `TII-252` ($n = 1{,}008, m = 10, t = 11, mt = 110$) [8] and Two-Anchor Holdout/Hermite to solve `TII-254` ($mt = 96$, reporting 27.2 GPU-hours for the two final Krylov sequences on 8× NVIDIA GH200) [9]. Algebraic vanishing attacks require almost no combinatorial search, yet their matrix dimensions grow steeply with Goppa degree t and code rate, leaving high-t instances such as `TII-228` ($t = 14$) through `TII-235` ($t = 16$) beyond current HOVER systems.

Primal-Shortened ISD [3] inverts that trade-off. Rather than exploiting full support, it shortens the code further to n' columns, harvests weight-$(2t + 1)$ codewords, and recovers (g, L) from quadratic polar forms. Puncturing no longer obstructs the attack, which explains why MiniMax's Route A reached `TII-228` ($mt = 126$, 12 punctures, 2,463 EPYC core-hours) and `TII-250` ($mt = 120$, 66,722 core-hours on 5,120–10,240 cores). Figure 1 and Table 1 place our five solved instances along this escalation path, showing why TII challenge IDs (estimated under 2023 attacks) no longer match Primal-Shortened ISD cost.

Table 1: Representative 2026 TII Track-2 key-recovery milestones and our five verified GPU recoveries.

Tier	Instance	(n, m, t)	mt	Method	Compute / Status
Prior	`TII-210/248`	$m = 9, t = 7$	63	Pfaffian / HOV [4, 5]	CPU (minutes)
Prior	`TII-253`	$(214, 8, 9)$	72	Hold-out / GIJS [6, 7]	CPU
Prior	`TII-254/252`	$m = 8, 10$	96, 110	2-Anchor / HOVER [8, 9]	27.2 GH200-h (`254`)
Control	`TII-245`	$(493, 9, 12)$	108	MiniMax / **Ours (1×L4)**	44.6 Xeon-ch vs. **19.6 m**
Benchmark	`TII-228`	$(500, 9, 14)$	126	MiniMax / **Ours (8×L40S)**	2,463 EPYC-ch vs. **2.15 GPU-h**
New	`TII-237`	$(500, 9, 15)$	135	**Ours (Colab L4)**	**22.1 L4-h** + 23.2 s lift
New	`TII-235`	$(502, 9, 16)$	144	**Ours (10×L40S)**	**39.62 L40S-h** + 18.0 s lift
New	`TII-185`	$(511, 9, 17)$	153	**Ours (`v5`: 7×L40S+3×L4)**	**140.4 L40S-h + 60 L4-h**
Margin	`348864+`	$(3488, 12, 64)$	≥ 768	Classic McEliece	Far beyond reach

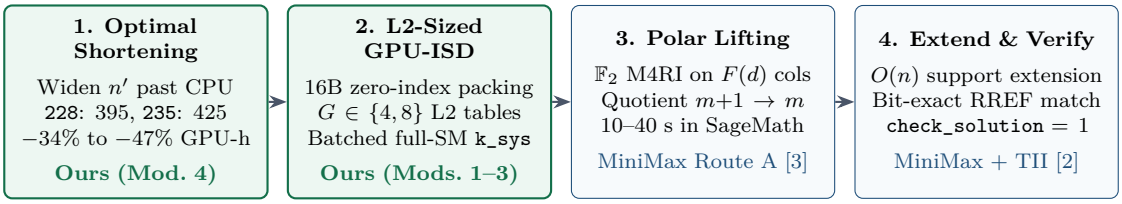


Figure 2: End-to-end Primal-Shortened ISD pipeline. Shaded green boxes mark our four GPU-oriented modifications; blue boxes reuse MiniMax's algebraic lift and the unmodified official TII verification harness.

3 Primal-Shortened ISD and Four GPU-Aware Modifications

Let $L = (\alpha_0, \ldots, \alpha_{n-1}) \in \mathbb{F}_{2^m}^n$ be an ordered support of n distinct elements and $g(X) \in \mathbb{F}_{2^m}[X]$ a monic square-free Goppa polynomial of degree t with $g(\alpha_i) \neq 0$ for all i. Taking a prefix $A = \{0, \ldots, n' - 1\}$ where $H_A \in \mathbb{F}_2^{mt \times n'}$ has full rank mt (verified empirically on every instance) yields the shortened Goppa code $\mathcal{C}_A = \ker(H_A) = \Gamma(L_A, g)$ of dimension $k' = n' - mt$. Any codeword $c \in \mathcal{C}_A$ of minimum odd Hamming weight $w = 2t + 1$ and support $S \subset A$ has locator polynomial $\sigma_S(X) = \prod_{i \in S}(X - \alpha_i)$ satisfying the characteristic-2 derivative-square identity $\sigma_S'(X) = g(X)^2$ (attributed to Finiasz in [3]). On the odd-weight affine slice of $\mathcal{C}_A$ (affine dimension $d = k' - 1$), every weight-$(2t + 1)$ codeword satisfies $m + 1$ quadratic relations over the $F(d) = 1 + d + \binom{d}{2}$ quadratic monomials in d coordinates, including the public quadratic weight identity $\sum_{i<j} c_i c_j = \binom{w}{2} \bmod 2$ with associated symmetric bilinear polar form T_{pub} [3]. When harvested words span an evaluation matrix of rank $F(d) - (m+1)$, dense $\mathbb{F}_2$ elimination recovers the $(m + 1)$-dimensional relation space. Modding out T_{pub} lifts the remaining m polar forms to an equivalent (L_A, g), and $O(n)$ field divisions then extend L_A to all n positions after applying the unscrambling transformation to H_{pub}.

Figure 2 separates MiniMax's upstream algebraic pipeline from our four GPU-oriented modifications.

MiniMax's Mode-6 CPU sampler performs full random systematic reduction followed by a Stern-style [10, 11] meet-in-the-middle search with $p = 3 + 3$ non-pivot bits, $l = 17$ collision bits ($2^{17} = 131{,}072$ buckets), and a 16 KB occupancy bitmap pre-filter. On a CPU, each thread owns one reduction whose hash table fits in L2/L3 cache. Moving that layout directly to a GPU breaks down: our per-SM port (`v1`: 58 blocks, 6 slots × 32 bytes per bucket, 25.7 MB/table) allocates 1.49 GB of live hash tables. Shrinking entries to 16 bytes with 4 slots and raising occupancy to 2 blocks/SM (`v2`: 116 blocks, 1.04 GB total) lowered `TII-237` throughput from 2,208 to 1,872 it/s because both allocations exceed the 50.3 MB L4 L2 cache by more than 20× while twice as many blocks contend for cache lines. Our hardened solver `src/minwords_gpu_v4.cu` (derived from MiniMax's `minwords_mt.c` under Apache-2.0) addresses this with four coupled modifications.

1. Lossless 16-byte packing without left indices. Because the bucket index $h \in [0, 2^{17})$ fixes the lowest 17 syndrome bits, any instance with $mt \leq 145$ packs all remaining $mt - 17 \leq 128$ bits losslessly into two 64-bit words:

$$p_0 = (s_0 \gg 17) \mid (s_1 \ll 47), \qquad p_1 = (s_1 \gg 17) \mid (s_2 \ll 47).$$

We store only (p_0, p_1) as a 16-byte `ulonglong2` and drop left-triple indices (a, b, c) from the hot bucket array. Under a uniform-syndrome heuristic given a 17-bit bucket collision, the conditional weight-filter pass probability $\binom{mt-17}{w-6}/2^{mt-17}$ is $2^{-27.02}$ on `TII-245`, $2^{-31.39}$ on `TII-228`, $2^{-33.57}$ on `TII-237`, and $2^{-35.74}$ on `TII-235`. Because the 128-bit packing is lossless up to $mt = 145$, every exact match $(\mathrm{popc}(p_0 \oplus r_{p_0}) + \mathrm{popc}(p_1 \oplus r_{p_1}) = w - 6)$ is a genuine weight-w codeword of the systematic matrix; only on these rare hits does the thread unpack (s_0, s_1, s_2) and recover (a, b, c) from shared-memory `colmask`. With 4 slots per bucket, each bucket payload is 64 bytes aligned (8.39 MB/table, plus a separate 0.52 MB count array and 16 KB bitmap). Under a Poisson load model, the 4-slot cap retains 98.03% of left triples on `TII-245` ($\lambda \approx 1.60$), 95.13% on `TII-237` ($\lambda \approx 2.20$), 89.4% on `TII-228` at $n' = 395$ ($\lambda \approx 2.99$), 85.8% on `TII-235` at $n' = 425$ ($\lambda \approx 3.41$), and 80.9% on `TII-185` at $n' = 447$ ($\lambda \approx 3.96$).

To cross the $mt = 145$ packing ceiling without enlarging the 16-byte bucket table, `src/minwords_gpu_v5.cu` raises `MAXR` $= 192$ while continuing to pack bits 17–144 into `ulonglong2`. For $145 < mt \leq 192$, the $e = mt - 145$ omitted upper bits can contribute at most e ones, so we apply the lossless range pre-filter $(w - 6) - e \leq \mathrm{popc}_{128} \leq w - 6$, reconstruct (a, b, c) from the exact 145-bit prefix, and evaluate the exact mt-bit weight against the full 3-word `colmask`. On `TII-185` ($mt = 153, e = 8, w = 35$), the range pre-filter passes with conditional probability $\sum_{j=21}^{29} \binom{128}{j}/2^{128} = 2^{-32.28}$ (vs. $2^{-37.91}$ for true hits), keeping false-positive reconstructions negligible while preserving the exact 35.7 MB / 71.4 MB L2-sized footprint.

2. Cooperative L2-sized execution ($G = 4$ on L4, $G = 8$ on L40S). Instead of assigning one table per SM, we allocate G concurrent tables whose combined footprint stays below the target GPU's nominal L2 cache. On NVIDIA L4 (58 SMs, 50.3 MB L2) and A100 (41.9 MB L2), $G = 4$ with 16 blocks per group (64 blocks total) uses 35,717,120 bytes (35.7 MB). On NVIDIA L40S (142 SMs, 100.7 MB L2), compiling with `-DG_GROUPS=8 -DBLOCKS_PER_GROUP=18` doubles concurrency to $G = 8$ (71.4 MB < 100.7 MB L2) and saturates all 142 SMs with 144 cooperative blocks. Every block caches its group's 16 KB occupancy bitmap in shared memory.

3. Batched systematic reduction and hardware scaling. Switching from `v2` to unbatched cooperative `v3` ($G = 4$) lifted `TII-237` throughput to only 2,671 it/s. CUDA event profiling over 256 iterations on L4 explained the ceiling:

`k_sys` : 68.00 ms (74.0%), `k_zero` : 0.34 ms, `k_build` : 10.82 ms, `k_probe` : 12.72 ms.

Running `k_sys<<<4, 256>>>` left 54 of 58 SMs idle during Gaussian elimination. Batching $64G$ independent reductions in one `k_sys<<<64G, 256>>>` launch (256 reductions / 2.2 MB state buffer on L4; 512 reductions / 4.4 MB on L40S) ahead of 64 build/probe steps raised L4 throughput 3.5× to 9,462 it/s on `TII-237` and 3.17× to 11,035 it/s peak (11,024 it/s full-run average) on `TII-245`. In our single-run `TII-237` comparison, an L4 (50.3 MB L2, 2.04 GHz boost, 1.94 CU/hr on Colab) beat an A100-SXM4-40GB (41.9 MB L2, 1.41 GHz boost, ≈ 6.8 CU/hr, 8,035 it/s) by 1.18× in raw speed and > 4× in iterations per CU. Scaling to L40S (100.7 MB L2, 142 SMs) reached 12,081 it/s on `TII-228` ($n' = 395$) and 10,934–15,593 it/s (aggregate 11,684 it/s) on `TII-235` ($n' = 425$).

4. GPU-aware optimal shortening window n'. MiniMax's `optimal_shortening_sizes` picks a narrow prefix n' whose random-code supply $N(n') = \binom{n'}{w}2^{-mt}$ exceeds the planning

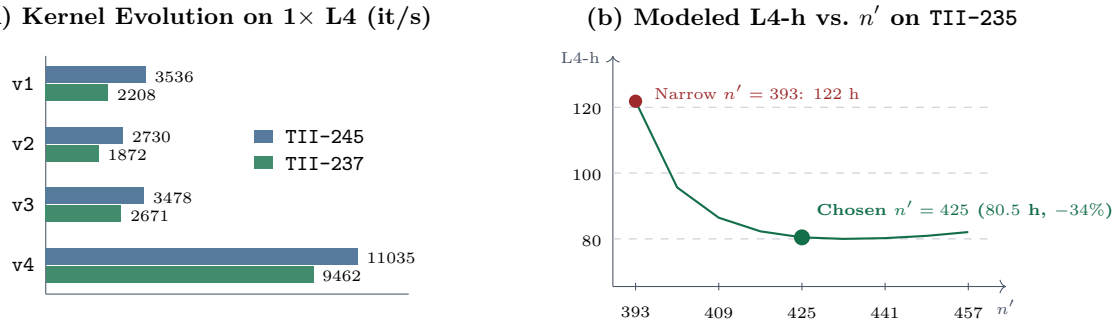


Figure 3: (a) Single-run `v1`–`v4` iteration throughput on one NVIDIA L4 (baseline sources in `src/baselines/`). (b) Modeled L4 GPU-hours vs. shortening prefix n' for `TII-235` ($mt = 144$), incorporating 4-slot Poisson bucket retention and coupon-collector depletion.

threshold $R(n') = F(k'-1) - (m+1)$ by roughly 1.4–2.2×, keeping the $\mathbb{F}_2$ linear system small. Under a Poisson-bucket and uniform coupon-collector model, expected iterations scale as

$$I(n') = \frac{-\ln\!\big(1 - 1.02\, R(n')/N(n')\big)}{\rho_4(\lambda)\, p_1(n')},$$

where $\rho_4(\lambda)$ is the 4-slot retention fraction at left-list load $\lambda = \binom{\lfloor k'/2 \rfloor}{3}/2^{17}$ and $p_1(n') = \binom{\lfloor k'/2 \rfloor}{3}\binom{\lceil k'/2 \rceil}{3}\binom{mt-17}{w-6}/\binom{n'}{w}$. Modeled GPU-hours equal $I(n') \cdot \tau(n')$, with per-iteration time $\tau(n')$ scaled in proportion to the left-list size $\binom{\lfloor k'/2 \rfloor}{3}$. A tight supply drives $-\ln(1 - R/N)$ sharply upward as duplicates accumulate. Widening n' grows $R(n')$ only quadratically (SageMath M4RI still finishes in under 40 s across all five instances) while $N(n')$ grows rapidly as $\binom{n'}{w}$, cutting duplicate waste and raising per-iteration yield until $\binom{k'/2}{3}$ and 4-slot overflow bend total GPU-hours back up (Figure 3, right). On `TII-228`, widening n' from a narrow $n' = 355$ point to $n' = 395$ lowers modeled L4 GPU-hours by 47% (7.4 h → 3.9 h; MiniMax's published `TII-228` run used $n' = 361$). On `TII-235`, widening n' from 393 to 425 lowers modeled L4 GPU-hours by 34% (122.0 h → 80.5 h), and in practice kept cross-worker overlap across 10 parallel GPUs to 2.06%.

4 Experiments

Figure 3 summarizes both the `v1`–`v4` kernel progression on L4 and the modeled shortening-window trade-off for `TII-235`. All five CUDA kernels (`v1`–`v5`) are published in our repository.

Control run: `TII-245` ($n = 493, m = 9, t = 12, mt = 108$). At MiniMax's prefix $n' = 326$, one NVIDIA L4 harvested 24,500 host-verified weight-25 codewords in 1,175.5 s (19.6 min, 12,958,464 iterations, 11,024 it/s average). SageMath 10.7 algebraic recovery on the Colab Xeon host completed in 9.78 s (1,185.3 s harvesting plus lift), returning exact quotient dimension 10, direct lift of dimension 9, bit-exact RREF equality ($H_{\text{cand}} = H_{\text{pub}}$), and `check_solution_track2.sage` (245) = 1. MiniMax [3] reported 2,500 s harvesting for 26,352 words and 2,510 s end-to-end (44.6 core-hours) on a 64-core Xeon Gold 6338 server (2.12× raw wall-clock ratio). Under uniform coupon-collector sampling with random-code supply $N = \binom{326}{25}/2^{108} \approx 52{,}299$, normalizing for their larger 26,352-word target gives a 1.91× speedup.

Head-to-head benchmark: `TII-228` ($n = 500, m = 9, t = 14, mt = 126$). MiniMax [3] solved `TII-228` at $n' = 361$ in 57.1 min wall-clock using 2,463 physical CPU core-hours across 2,048–5,120 AMD EPYC cores. Using our wider prefix $n' = 395$ ($\operatorname{rank}(H_A) = 126$, $k' = 269$, threshold 36,037), we ran eight Modal NVIDIA L40S workers ($G = 8$, 144 blocks, distinct seeds)

from scratch, each targeting 4,720 weight-29 codewords. Every worker finished with zero output-buffer overflows in 949.6–989.2 s (12,025–12,125 it/s, mean 12,081 it/s), accumulating 93,574,144 exact ISD iterations across 2.15 L40S GPU-hours. End-to-end wall-clock time including Modal container startup and checkpoint transfer was 18.90 min (≈ $4.30). Deduplicating the 37,760 raw words yielded 37,084 unique codewords (1.79% cross-worker overlap). SageMath 10.5 on an 8-core CPU container completed the algebraic lift in 24.64 s (19.94 s elimination, quotient dim 10, direct lift dim 9) and passed `check_solution_track2.sage` (228) = 1 in 5.43 s. Although the two runs used different shortening prefixes and distinct hardware cost models (CPU core-hours vs. GPU-hours), eight L40S GPUs cut wall-clock recovery time on `TII-228` by 3.0× relative to MiniMax's cluster run.

Codimension milestone: `TII-237` ($n = 500, m = 9, t = 15, mt = 135$). At $n' = 378$ ($\text{rank}(H_A) = 135$, $k' = 243$, threshold 29,394, random-code supply ≈ 62,943), we harvested 29,742 unique weight-31 codewords across Colab L4 workers totaling ≈ 22.1 GPU-hours and $\approx 7.4 \times 10^8$ iterations (708.6M measured across five logged L4 workers plus ≈ 30M estimated on three short exploratory runs; full per-worker ledger in `codewords/RUN_MANIFEST.md`). Capture–recapture between our two largest independent-seed L4 workers ($|A| = 17{,}009$, $|B| = 8{,}570$, $|A \cap B| = 2{,}215$) gave $\hat{N} = 65{,}809 \pm 1{,}123$ (95% CI $[63{,}608, 68{,}010]$), within 5% of the random-code prediction. SageMath 10.7 lifted the key in 23.23 s (15.33 s elimination, quotient dim 10, direct lift dim 9) and passed `check_solution_track2.sage` (237) = 1. Total Colab spend across all development, profiling, `TII-245`, and `TII-237` was 68.97 CU.

Approaching the 128-bit packing ceiling: `TII-235` ($n = 502, m = 9, t = 16, mt = 144$). Punctured at 10 positions ($512 - 502 = 10$) with Goppa degree $t = 16$, `TII-235` sits well beyond current vanishing-attack parameters and requires $144 - 17 = 127$ packed syndrome bits—within one bit of our 128-bit `ulonglong2` limit. At our widened prefix $n' = 425$ ($\text{rank}(H_A) = 144$, $k' = 281$, threshold 39,331, random-code supply ≈ 787,006), ten Modal NVIDIA L40S workers ($G = 8$, 144 blocks, distinct seeds) each targeted 4,120 weight-33 codewords. All ten workers completed with zero overflows across 1.67×10^9 exact iterations and 39.62 L40S GPU-hours in 4.30 h wall-clock (≈ $79; Table 2). Merging the 41,200 raw words produced 40,353 unique codewords (2.06% overlap). Despite the widened 39,341-feature matrix, SageMath 10.5 completed the algebraic lift in 17.97 s (15.06 s elimination, exact quotient dim 10, direct lift dim 9), recovering a monic irreducible degree-16 polynomial $g(X) \in \mathbb{F}_{2^9}[X]$ and 502-element support L with bit-exact RREF match and `check_solution_track2.sage` (235) = 1 in 3.37 s.

Crossing the 128-bit ceiling: `TII-185` ($n = 511, m = 9, t = 17, mt = 153$). At $mt = 153$, `TII-185` exceeds our lossless 128-bit packing format by 8 bits and tests `src/minwords_gpu_v5.cu` directly. At widened prefix $n' = 447$ ($\text{rank}(H_A) = 153$, $k' = 294$, $F(293) = 43{,}072$, threshold 43,062, random-code supply ≈ 1,245,602), we ran a hybrid 10-GPU fleet combining seven Modal L40S workers ($G = 8$, 144 blocks, target 5,600 each) and three Colab L4 workers ($G = 4$, 64 blocks). Across 21.1 h wall-clock on Modal (140.4 L40S GPU-hours, ≈ $281, 5.21×10^9 exact iterations, 35,726 raw words) and ≈ 60.0 L4 GPU-hours on Colab (1.17×10^9 exact iterations, 8,031 raw words), the fleet accumulated 6.38×10^9 exact ISD iterations and 43,757 raw weight-35 codewords, deduplicating to 43,117 unique codewords (1.46% cross-worker overlap; Table 2). SageMath 10.5 on an 8-core CPU container completed the algebraic lift in 39.01 s (31.60 s elimination, exact quotient dim 10, direct lift dim 9), recovering a monic irreducible degree-17 polynomial $g(X) \in \mathbb{F}_{2^9}[X]$ and 511-element support L with bit-exact RREF match and `check_solution_track2.sage` (185) = 1 in 6.12 s. Table 3 records all fifteen SHA-256 verification digests.

Table 2: Summary of all five verified GPU key recoveries. Total iterations for `TII-245`, `TII-228`, `TII-235`, and `TII-185` are exact device counters; `TII-237` combines 708.6M measured and ≈ 30M estimated iterations. For `TII-185`, wall-clock (21.1 h) reports the Modal fleet duration (22.2 h including the Colab head start).

Instance	mt	n'	Hardware	Wall	GPU-h	Iterations	Unique Words	Lift (s)
`TII-245`	108	326	1× Colab L4	19.6 m	0.33	12.96M	24,500	9.78
`TII-228`	126	395	8× Modal L40S	18.9 m	2.15	93.57M	37,084	24.64
`TII-237`	135	378	Colab L4 workers	—	≈ 22.1	$\approx 7.4 \times 10^8$	29,742	23.23
`TII-235`	144	425	10× Modal L40S	4.30 h	39.62	1.67×10^9	40,353	17.97
`TII-185`	153	447	7×L40S + 3×L4	21.1 h	140.4+≈60	6.38×10^9	43,117	39.01

Table 3: Cryptographic verification digests ($\mathbb{F}_{2^9} = \mathbb{F}_2[a]/(a^9 + a^4 + 1)$). All five equivalent private keys pass the official TII SageMath checker; our pure-Python verifier additionally verifies all 174,796 published codewords and SHA-256 digests.

Instance	Artifact	SHA-256 Digest
`TII-245`	`pk_McEliece_245.txt`	`ea029e86d34d13334bb9439f87333a8e87ebd902379ec515331d2bed78e973a8`
	`sk_McEliece_245.txt`	`967d2f4e934ca7617bdd71cfb5a0fcd98ec0f54a2c50db5170899ab3debac523`
	`words_245_np326.txt.gz` (24,500)	`b54b11b6c695b32a08aedb545cfb68dac9e0e3e7e1184175474344bfc59b4a11`
`TII-228`	`pk_McEliece_228.txt`	`f98ddc98e6f54a60308b49ec36e16f70b6249ba4b7eac79f72cdf7fe70649f5a`
	`sk_McEliece_228.txt`	`717388cc390977ad536e23bdb6c7bc8e2cf69f065d69e3133f7fd27efd6dd51d`
	`words_228_np395.txt.gz` (37,084)	`46760cda2c768cdf3be5a559ed6a65895afe68d9e7241f52b91c80aede1e93d8`
`TII-237`	`pk_McEliece_237.txt`	`b3cba630833e00d8e225d1943be87c31ddd74f270bd73f61c373aa70b1356f3b`
	`sk_McEliece_237.txt`	`128e26137b2b2e8128316cbc5e1c7766073d790e26a308a9766a120f05770e0f`
	`words_237_np378.txt.gz` (29,742)	`7793b74438d76235c8f5048a6b5644113b8372d999f837e1c09c274b2a472531`
`TII-235`	`pk_McEliece_235.txt`	`f56256c7b6f253ff7748207ee05545d92644f6f8c5a7189fcaff9442eedd87a5`
	`sk_McEliece_235.txt`	`e96fdd90a9364e275d110dfb20b3fc56553123a7fb9d65e4c64772dbc195c221`
	`words_235_np425.txt.gz` (40,353)	`e8ee29c0b4750d3da4ac48b9e27d99c9c14598dad46bcf6526cd093e35d5329c`
`TII-185`	`pk_McEliece_185.txt`	`19d1b2d4425b0541bb2f9061f3d3271c69159b95ce961014fefd79499293b140`
	`sk_McEliece_185.txt`	`adf36e4b00a35549e887a25793e2361b0ea53dbbae1ec888b7317a8532f1a9db`
	`words_185_np447.txt.gz` (43,117)	`5c9a58e7361707792d2b0e72c807238176049e545524c9c0d6b38af4e5670e55`

5 Limitations and Artifacts

Cache residency and hardware variance. Sizing the table footprint below nominal L2 capacity (35.7 MB on L4, 71.4 MB on L40S) does not prove 100% L2 hit rates without Nsight hardware counters, and the `v1`–`v3` and A100 comparisons are single-run observations without $G \in \{2, 4, 8\}$ ablations on identical hardware. On Modal, L40S nodes exhibited two distinct boost-clock tiers (11.0k vs. 15.6k it/s on `TII-235`), which accounts for the 3.0–4.3 h spread across workers.

Sampling bias and non-determinism. Non-pivot columns are partitioned in sorted order rather than shuffled per reduction, introducing sampling bias; 4-slot buckets discard left triples in atomic arrival order (≈ 4.9% on `TII-237`, ≈ 14.2% on `TII-235`, and ≈ 19.1% on `TII-185`), making per-iteration output non-deterministic. Both effects render capture–recapture supply estimates heuristic.

Domain bounds and key equivalence. `src/minwords_gpu_v4.cu` enforces $mt \leq 145$ (lossless) and `src/minwords_gpu_v5.cu` extends to $mt \leq 192$ via truncated 16-byte range pre-filtering, both with $k' \leq 320$ and `mode` $= 6$. Moving past $mt = 153$ hits either $m = 9, t = 18$ ($mt = 162$) or $m = 10, t = 16$ ($mt = 160$, $k' \approx 397$), where modeled ISD search cost climbs by 5× to $> 100\times$. All recovered witnesses certify *equivalent* private keys (g, L) generating H_{pub} bit-for-bit, not necessarily the originally sampled generator representation.

Artifacts. All CUDA sources (`v1`–`v5`), compressed codeword datasets (174,796 verified words across five instances), equivalent private keys, and run manifests are available at https://github.com/raullenchai/tii-mceliece-gpu under Apache-2.0 (preserving MiniMax's notices in `LICENSE` and `NOTICE`). Running

```
python3 src/verify_witnesses.py --check-words
```

checks all five equivalent private keys, hardcoded SHA-256 digests, and all 174,796 harvested codewords in pure Python with zero external dependencies, while `src/solve_from_words.py` reproduces the SageMath algebraic recovery directly from the published codeword archives.

6 Conclusion

Packing MiniMax's Mode-6 Stern harvester into the GPU L2 cache, widening the shortening window n', and extending past $mt = 145$ via truncated 16-byte range pre-filtering moves high-codimension Primal-Shortened ISD from multi-thousand-core CPU clusters to a handful of serverless GPUs. Eight Modal L40S GPUs solve `TII-228` ($mt = 126$) in 18.9 min (2.15 GPU-hours), Colab L4 workers solve `TII-237` ($mt = 135$) in $\approx$ 22.1 GPU-hours, ten Modal L40S GPUs recover `TII-235` ($mt = 144$) in 4.30 h (39.62 GPU-hours, $\approx$ \$79), and a hybrid 7$\times$ L40S + 3$\times$ L4 fleet recovers `TII-185` ($mt = 153$) in 21.1 h Modal wall-clock (140.4 L40S + $\approx$ 60 L4 GPU-hours, 39.01 s lift). These experiments do not demonstrate practical key recovery at proposed Classic McEliece parameter sets ($m \geq 12, t \geq 64, mt \geq 768$).

Acknowledgments

We thank Modal (https://modal.com) for providing serverless NVIDIA L40S GPU infrastructure for our `TII-228`, `TII-235`, and `TII-185` experiments, Google for providing NVIDIA L4 and A100 GPU compute through Google Colab for our `TII-245`, `TII-237`, `TII-185`, and kernel-profiling experiments, and MiniMax for releasing their Primal-Shortened ISD framework and Mode-6 sampler under Apache-2.0.

References

[1] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," *DSN Progress Report*, 42-44, pp. 114–116, 1978.

[2] E. Kirshanova and A. Esser, *TII Decoding Challenges*, https://github.com/ElenaKirshanova/tii_decoding_challenge.

[3] MiniMax, "Primal-Shortened ISD Key Recovery on Binary Goppa Codes," https://github.com/MiniMax-AI/Research/tree/main/Cryptography/TII-McEliece, Nov. 20, 2026.

[4] M. Bardet, A. Lemoine, and J.-P. Tillich, "An attack on the CFS scheme and on TII McEliece challenges," *IACR ePrint 2026/430* (CRYPTO 2026), https://eprint.iacr.org/2026/430.

[5] T. Hemmert, "Key recovery for the McEliece cryptosystem using higher-order vanishing," *IACR ePrint 2026/1339*, 2026, https://eprint.iacr.org/2026/1339.

[6] K. Vedenev, "Extending distinguishing to key recovery for subfield subcodes of GRS codes," *IACR ePrint 2026/1747*, 2026, https://eprint.iacr.org/2026/1747.

[7] S. A. Weis, "Improving GIJS key recovery for Classic McEliece," *IACR ePrint 2026/1984*, 2026, https://eprint.iacr.org/2026/1984.

[8] M.-J. O. Saarinen, "HOVER: Higher-order vanishing endomorphism recovery," *IACR ePrint 2026/1778*, 2026, https://eprint.iacr.org/2026/1778.

[9] M.-J. O. Saarinen, "Two-Anchor Holdout/Hermite: Solving the TII-254 McEliece key recovery challenge," *IACR ePrint 2026/1986*, 2026, https://eprint.iacr.org/2026/1986.

[10] J. Stern, "A method for finding codewords of small weight," in *Coding Theory and Applications*, LNCS 388, pp. 106–113, Springer, 1989.

[11] I. Dumer, "On minimum distance decoding of linear codes," in *Proc. 5th Joint Soviet–Swedish Int. Workshop on Information Theory*, pp. 50–52, 1991.