On Memory Effects in PWXL variants

Jintai Ding Hao Guo Bo-Yin Yang

August 2, 2026

Abstract

We estimate the intrinsic undercounting in the free-memory-access, Macaulay coefficient-on-demand RAM modeling when applied to the Parallelized Wiedemann-based XL in the Ran Wedge attack and in the Furue–Ikematsu intersection attack, under some optimistic but still feasible-sounding assumptions for the attackers. We believe that this shows the memory effects makes UOV secure enough for Ip, Is, and III. If NIST considers our original parameters insufficiently convincing, we do not take Furue's suggested replacements; we offer instead the following perturbations, which hold m — and hence the compressed public key — fixed and spend only on the vinegar count: uov-Ip# (256,116,44), uov-III# (256,186,72) and uov-V# (256,250,96).

Synopsis

Parallelized Wiedemann-based XL (PWXL) is the current state of the art for several cryptanalytic attacks, among them the cryptanalysis of UOV — where the best attacks are a version of Ward Beullens' intersection attack and, in characteristic 2, the Ran wedge-product attack. It is sometimes claimed that the Crypto 2022 paper (Baena–Briaud–Cabarcas–Perlner–Smith-Tone–Verbel, `https://eprint.iacr.org/2021/1677.pdf`) showed how to make the memory-access cost of the PWXL stage negligible. We argue it does not, for the attacks that actually matter against UOV. Two framing points first:

- The Crypto 2022 paper works mostly on the *rectangular MinRank* attack, a bipartite XL at bi-degree $(b, 1)$. The natural subdivision it exploits is over the *minor* variables. In current UOV cryptanalysis (intersection and wedge attacks) the key step is a *regular XL*, where no such subdivision exists.
- Because the hash that partitions memory is a *function*, a row's monomials are distributed across the clusters it reaches rather than replicated into them. This decides which costs the fan-out multiplies and which it does not, and it is the single most important bookkeeping point in what follows.

The claims we will defend. (i) The per-row fan-out — the number of processing clusters that must fire — is $\Theta(h^2)$, not $\Theta(h)$, for a quadratic seam; this is a genuine structural difference from Support-Minors, and it governs communication and row-selection cost, but *not* per-monomial address generation. (ii) The keep-rate and the shared-support reuse factor are not independent: $n'_r = \min(m, mT/R)$ — the identity $(R/T) \cdot n'_r = m$ in the overdetermined regime, the cap

$n'_r = m$ below it — so one cannot posit both a tall-thin matrix and an absence of per-block reuse. At the operating point of these attacks the identity forces $R/T = \mathrm{O}(1)$ and $n'_r = 16$–88. (iii) The on-the-fly cost remains 9–14 bits above the "sub-0.1%" of Appendix B.1, and the "embarrassingly parallel" argument still fails for regular XL. (iv) Pricing memory access defensibly lifts Ran's wedge attack [ePrint 2025/1143] above the NIST gate thresholds on every affected set, by +14 to +61 bits, and comfortably so even in the physically over-generous 3-D model. (v) Against the more aggressive Furue–Ikematsu intersection attack [ePrint 2026/298] the same accounting clears uov-Is and — barely — uov-III, but leaves uov-Ip and uov-V below threshold. We flag that gap rather than paper over it. Crucially, it is the memory term that does this work: on-the-fly cost alone no longer clears level I even against Ran.

1 Historical comment on a comment

On p. 397 of "Improving Support-Minors Rank Attacks," the authors write that *the cost of distributing the MinRank instance and a PRNG seed to the 2^h processing clusters ... is insignificant.* They cite [Baena et al., cites 16 and 40] for this. But in Cheng–Chou–Niederhagen–Yang [cite 16], the authors say only that entries can be generated on demand with negligible *memory*, immediately adding that *the computation of the entry positions requires additional time*; and Ding et al. [cite 40] explicitly flag that recreating the matrix on the fly *underestimates the complexity.* Neither conceded that the cost is negligible. We argue it is not.

2 Building (and using) the matrix on the fly

Building a random matrix is not the problem; *using* it is.

2.1 Shared-support rows, and an identity that fixes the reuse factor

Both Support-Minors and the intersection/wedge attacks have blocks of equations with shared support sets, which has been used to argue positions can be cached. One might argue the opposite: that the Macaulay matrix is tall and thin, that squaring it up by random row-dropping leaves only $n'_r = \mathrm{rank}/\#\mathrm{blocks} = \mathrm{O}(1)$ survivors per block, and hence that a position table written once is essentially never re-read. That argument does not survive a closer look, and the reason is worth stating as a proposition because it also settles Section 3.

Let R be the candidate rows, $T = V$ the columns — equivalently the rows kept after squaring up — and n'_r the survivors per shared-support block. A block is $\{\mu \cdot f_i : i = 1..m\}$ for a fixed multiplier μ: every f_i is a dense quadratic in the same coordinates, so all m rows have the same monomial support. Each block therefore holds exactly m rows and $\#\mathrm{blocks} = R/m$, whence

Proposition. *$n'_r = T/\#blocks = mT/R$ whenever $R \geq T$, that is $(R/T) \cdot n'_r = m$; when $R < T$ every candidate row is kept and n'_r caps at the block height, so in general $n'_r = \min(m, mT/R)$.*

Three of the four intersection sets sit below $R/T = 1$ and take the cap — which is why the table below shows $n'_r = m$ there while the wedge rows satisfy the identity exactly.

A large keep-rate and an absence of per-block reuse are the same statement read twice, and their product is pinned at m. Both are governed by D/N_{gs}. For squarefree columns

$$R/T \;=\; m\binom{N_{\mathrm{gs}}}{D-2}\Big/\binom{N_{\mathrm{gs}}}{D} \;=\; \frac{m\,D(D-1)}{(N_{\mathrm{gs}}-D+2)(N_{\mathrm{gs}}-D+1)},$$

which crosses 1 at the binomial midpoint $D/N_{\mathrm{gs}} = 1/2$ and rises steeply past it. The near-determined regime familiar from the XL experiments of Chou, Niederhagen et al. (CHES 2012; Niederhagen thesis, §4.5.3) — m only slightly above n, many syzygies — has $D_{reg}/n > 1/2$ and hence $R/T = 2^4 \,..\, 2^7$ with $n'_r = \mathrm{O}(1)$. That is where the tall-thin intuition comes from, and for those systems it is correct. For $n = 16$, $m = 18$ the series $(1-t)(1+t)^{18}$ first goes non-positive at $d = 10$, so $D/n = 0.625$, and at $d = 9$ one has $\binom{16}{7} = \binom{16}{9}$ exactly, giving $R/T = m = 18$. The attacks under discussion are in the other regime:

	N_{gs}	D	D/N_{gs}	R/T	n'_r
wedge uov-Ip	86	18	0.209	2.79	15.8
wedge uov-Is	118	22	0.186	3.11	20.6
wedge uov-III	138	26	0.188	3.63	19.8
wedge uov-V	180	32	0.178	4.26	22.5
int. uov-Ip	105	13	0.124	0.79	44.0
int. uov-Is	144	16	0.111	0.92	64.0
int. uov-III	170	18	0.106	0.94	72.0
int. uov-V	222	22	0.099	1.09	87.9

So 16 to 88 rows per block survive, and the reuse those rows permit is real: a cluster computes the column indices for its share of the block's support and caches the corresponding vector coordinates once — a 2 to 8 KB local table — then reuses both across every survivor. This is exactly the amortization Baena et al. invoke to set $\rho_m = 1/n'_r$, and consistency demands the same $1/n'_r$ be applied to address generation. We therefore grant it on both terms. This costs us about 4 bits on the wedge figures and 5.5 to 6.4 on the intersection figures, and we would rather concede it than defend an assumption the identity contradicts.

2.2 A minimalist cost per address generation

Appendix B.1 of the Crypto 2022 article puts the extra work of building the Macaulay matrix on the fly at less than 2^7 gates per multiplication. This neglects that for each term one must recompute the canonical index of a squarefree monomial from scratch. Colex-ranking a size-D subset $s_1 < \cdots < s_D$ of $[N_{\mathrm{gs}}]$ to its index is $\sum_{j=1}^{D} C(s_j, j)$ — D cached-table lookups and D-1 additions — and *un*ranking an index back to the coordinate it addresses is a greedy descent of the same length, one compare-and-subtract per position. Either way the work is $\Theta(D)$ L-bit adds, not a fixed handful. Adding L-bit indices is expensive in gates relative to a binary-field multiply: a single L-bit NAND/NOR add costs 9L-3 gates.

One refinement, in the attacker's favor, before we tabulate. Charging all D additions at the full width $L = \lceil \lg \binom{N_{\mathrm{gs}}}{D} \rceil$ is loose: colex ranking accumulates from the small end, and for $\{s_1 < \cdots < s_D\}$ a subset of $\{0, \ldots, N_{\mathrm{gs}} - 1\}$, maximizing the first j terms must still leave room for $s_{j+1} < \cdots < s_D \le N_{\mathrm{gs}} - 1$, forcing $s_i \le N_{\mathrm{gs}} - 1 - D + i$. The partial sum after j terms is therefore at most — and exactly attains —

$$S_j \;=\; \sum_{i=1}^{j} \binom{N_{\mathrm{gs}} - 1 - D + i}{i},$$

realized by the top subset $\{N_{\text{gs}} - D, \ldots, N_{\text{gs}} - 1\}$. The j-th add thus needs exactly $\lceil \lg(S_j + 1) \rceil$ bits, with no slack; we have verified the bound by exhaustive enumeration over all D-subsets for small (N_{gs}, D). The honest per-monomial cost is

$$\sum_{j=1}^{D} \big(9\lceil \lg S_j \rceil - 3\big), \quad \text{rather than } D(9L - 3).$$

Because the early terms are tiny and $\lceil \lg S_j \rceil$ climbs to L only near $j = D$, the mean add-width is 0.600 to 0.612 of L across our parameter range; we quote **0.61**. This cuts the *attacker's* cost, so we record the flat $D(9L - 3)$ alongside for transparency:

	uov-Ip (wedge)	uov-Ip (int.)	uov-III (wedge)	uov-III (int.)	uov-V (wedge)	uov-V (int.)
dimension	$2^{60.4}$	$2^{53.6}$	$2^{92.8}$	$2^{79.5}$	$2^{117.8}$	$2^{100.0}$
index width L	61	54	93	80	118	100
one L-bit add = 9L-3	$2^{9.1}$	$2^{8.9}$	$2^{9.7}$	$2^{9.5}$	$2^{10.0}$	$2^{9.8}$
subset size D	18	13	26	18	32	22
flat $D(9L-3)$	$2^{13.3}$	$2^{12.6}$	$2^{14.4}$	$2^{13.7}$	$2^{15.0}$	$2^{14.3}$
tightened, 0.61	$2^{12.55}$	$2^{11.90}$	$2^{13.68}$	$2^{12.91}$	$2^{14.31}$	$2^{13.52}$

So address generation costs 2^{12} – 2^{14} gates per monomial. Against the paper's own field-multiply cost $2(\log_2 q)^2 + \log_2 q = 2 \cdot 8^2 + 8 = 136 \approx 2^{7.09}$ gates, that is a $2^{4.8}$ to $2^{8.0}$ tax **per monomial before any amortization**, and $2^{-0.7}$ to $2^{3.7}$ after the $1/n'_r$ reuse of Section 2.1 (the wedge occupying the upper end, $2^{1.5}$ to $2^{3.7}$, the intersection the lower) — against the $\approx 2^{-10}$ asserted in Appendix B.1, still 9 to 14 bits too optimistic.

Why one generation per monomial, and why batching fails. Given the cached colex rank of μ, whose support A has size D-2, appending the quadratic's two indices $\{i, j\}$ costs two adds instead of D only if both exceed max(A): then the merged sequence is $a_1 < \cdots < a_{D-2} < i < j$ and the first $D - 2$ terms are unchanged. Any interleaving moves some a_k from position k to k+1, changing its term from $\binom{a_k}{k}$ to $\binom{a_k}{k+1}$ and shifting every higher term — a full recomputation. Batching therefore applies exactly when $\{i, j\}$ are the top two of the final D-set, a fraction $1/\binom{D}{2}$ of monomials: 0.654%, 0.433%, 0.308%, 0.202% for the wedge sets, worth under 0.02 bit. The same event is the only one on which incremental in-place index advance yields sequential ranks, which disposes of that objection too.

2.3 Partitioning cost in the favorable case of Support-Minors

If one *could* split the minor-variable space so each equation is computed at a handful of nodes — say 2^3 nodes for Support-Minors — the attacker has spent $2^3 \times 2^{11} \approx 2^{14}$ to $2^3 \times 2^{13} \approx 2^{16}$ gates per matrix element. Even in this most-favorable case it is no longer 0.1% of the smallest error term.

2.4 Why intersection / wedge attacks are worse: the fan-out is quadratic in h

Here is the structural point. Hash a monomial by its support intersected with the first h variables, $H(S) = S \cap [h]$. A regular-XL row is $\mu \cdot f$ (or $\mu \wedge f$) with f a **dense quadratic**; its

monomials are $A \cup \{i, j\}$ with $A = supp(\mu)$. As $\{i, j\}$ ranges over the quadratic, the hash ranges over A', A'+one, A'+two ($A' = A \cap [h]$) — a **radius-2 ball** in the first h coordinates, of size

$$1 + (h - a) + \binom{h-a}{2} \approx 1 + h + \binom{h}{2} = \Theta(h^2).$$

This is the whole difference from Support-Minors, where every monomial carries a minor variable c_T with T a d-subset of one *fixed* $(d+1)$-set J, so only $d+1$ hash patterns are reachable — a linear, $h(d+1)/n_c$ fan-out. There is no fixed small J to confine a subset's support, so the ball replaces the star. At $h = 11$ with $a \approx 2$ the ball has $g = 1 + 9 + \binom{9}{2} = \mathbf{46}$ clusters, and it stays near 2^6 across the realistic range $h = 11$–16, independent of the parameter set and not shrinking at higher solving degree.

What the fan-out multiplies, and what it does not. It is tempting to charge address generation at every firing cluster, giving an overhead of fan-out × addr per useful multiply. That is wrong. The hash H is a *function*, so each monomial lands in exactly one bucket: a row's w monomials are **partitioned** across the ball, not replicated into it, and the total work is w independent of g. A cluster owning bucket S enumerates its share in closed form from (A, S), driven by $|S \quad A'|$: two forced indices give exactly one monomial; one forced index gives about $N_{\text{gs}} - h$; and $S = A'$ gives about $C(N_{\text{gs}} - h,\ 2)$. The mass is lopsided, with $m_{gs} = v+2$ the ground set of the wedge quadratic:

set	m_{gs}	w	r=0 (1 bucket)	r=1 (9 buckets)	r=2 (36 buckets)
uov-Ip	70	2278	1711 = 75.1%	531 = 23.3%	36 = 1.6%
uov-Is	98	4560	3741 = 82.0%	783 = 17.2%	36 = 0.8%
uov-III	114	6216	5253 = 84.5%	927 = 14.9%	36 = 0.6%
uov-V	150	10878	9591 = 88.2%	1251 = 11.5%	36 = 0.3%

The $\binom{h-a}{2} = 36$ top-shell buckets that *generate* the $\Theta(h^2)$ count hold one monomial each — under 2% of the row between them — while about 85% sits in the single bucket $S = A'$. Charging every cluster a full row's worth of addressing would overcount by $\log_2(g) = 5.5$ bits, and would do so by counting precisely the emptiest buckets in the ball. Accordingly:

- **address generation** is charged once per monomial, with no fan-out factor;
- **row selection by PRF** does carry $(1 + g)$, because it is a per-(row, cluster) *decision* and every firing cluster must evaluate the PRF to learn whether the row is kept — this is Baena et al.'s own count [1677, App. B.2, where the PRF is evaluated $1 + h(d+1)/n_c$ times per row];
- **memory access** is per-access and carries neither.

The $\Theta(h^2)$ fan-out therefore remains a real and novel obstruction relative to Support-Minors, but as a statement about communication and coordination: 46 partial sums per row must be routed and combined. That is what defeats "embarrassingly parallel," and Section 7 shows it is also what makes h impossible to raise in hardware. It is not a multiplier on gate-level address arithmetic.

Subdividing by degree instead does partition the monomials, but almost the entire monomial mass sits at the top degree, so it does not help.

2.5 Row weight and the Ran encoding

For the wedge attack the row weight is $w = \binom{v+2}{2}$ (the density of the pruned wedge matrix in the UOV round-2 analysis), **not** $\binom{n}{2}$: the alternating polar form q_i couples only through the vinegar-indexed part, so $\binom{v+2}{2}$ is correct, not a typo. That gives $w = 2^{11.2}/2^{12.2}/2^{12.7}/2^{13.4}$ for Ip/Is/III/V.

Combining the per-monomial address cost with the $1/n'_r$ reuse of Section 2.1, the overhead per *useful* field multiply is

$$\frac{2^{12}\text{–}2^{14}\ (\Theta(D)\ \text{addr/monomial})}{\left(16\text{–}23 : n'_r\right)\left(2^{7.09} : \mathbb{F}_q\ \text{mult}\right)} \approx 2^{1.5}\text{–}2^{3.7}\times,$$

that is, address generation adds 1.9 to 3.8 bits on top of the RAM count for the wedge sets. It remains 11 to 14 bits above the sub-0.1% ($\approx 2^{-10}$) claimed in Appendix B.1, so the negligibility claim does not survive; but the on-the-fly term is a backstop of a couple of bits, not a decisive one, and the security margins below rest on the memory term.

3 On doing things cheaply by random seed

The Baena et al. costing assigns one AES call (2^{16} NAND gates) per node per candidate row. Per useful multiply this is AES$\cdot(R/T)\cdot(1+g)/w$. Taking the keep-rate ratio R/T to be 2^7 would — by the identity of Section 2.1 — force $n'_r < 1$; that value is characteristic of the near-determined XL regime, not of these attacks. At the true $R/T = 0.79$ to 4.26 the seed term is $2^{7.1}$ to $2^{11.8}$ gates per multiply, that is 2^0 to 2^6 times a field multiply. As *described* this is not "insignificant," but per useful multiply it **exceeds** address generation at every parameter set, by 0.03 to 3.22 bits — FLY is divided by $n'_r = 16$–88 while RNG already carries n'_r inside R/T — so the seed step as described is the larger of the two on-the-fly terms. For the intersection attack R/T sits at or just above 1 (0.79/0.92/0.94/1.09), so essentially every generated equation is used and the selection machinery is not merely cheap but very nearly absent.

A cheaper row generator (pending empirical validation). Partition by the index-set part of the original-row address so each block corresponds to a few related monomials. Pick a small integer μ and take μ or $\mu-1$ rows from each block according to whether the block index j (already in (multi)set form) crosses a threshold; a single PRF call keyed by the (multi)set index decides the picks for the whole block. Advancing a (multi)set index in place is far cheaper than converting to an integer, incrementing, and converting back. This spends one call per *picked* equation rather than per node per candidate row. Cryptography remains an empirical science: anyone taking this route needs mid-range working examples to confirm the generated systems are valid and solvable.

3.1 Squeezing harder — many "random" picks from one call

The costing above charges one full PRF call per selected row. That is wasteful: a row selection is a tiny amount of entropy, and a single primitive call produces far more output than one pick consumes. The natural optimization is to *squeeze*: draw one wide block of pseudorandom bits

per call and carve it into many independent picks, advancing the (multi)set index in place rather than re-keying.

How many bits per pick. The universe is $N \in [2^6, 2^8]$. A single choice out of N carries $\lceil \log_2 N \rceil = 6$–$8$ information bits; an unordered distinct pair carries $\log_2 \binom{N}{2} \approx 2 \log_2 N$ - $1 = 11$–15. Reducing b drawn bits mod N leaves a bias $\leq N/2^b = 2^{-g}$ with $g = b - \log_2 N$ guard bits. Crucially the standard here is *solvability* — the selected square submatrix must be full rank with high probability — not indistinguishability, so a per-pick bias of 2^{-8} is already ample and $g \approx 8$ suffices (against a distinguisher one would need $g \approx 64$–128; the attack does not). This gives $\sim$14–16 bits for a single pick and $\sim$28–32 bits for a pair drawn as two indices with collision-rejection. Naming the pair canonically would cost only $\sim$19–23 bits, but that reintroduces the combinatorial unrank — precisely the $\sim 2^{13}$ gates of address arithmetic we are trying to charge — so a gate-minimizing attacker draws two indices and eats the $\sim$30 bits. For the stated mix (mostly singles, occasional pair) budget $\sim$20–24 bits/pick.

Cost per pick after squeezing. The table gives the per-bit keystream cost of candidate primitives and the resulting picks-per-call and gates-per-pick at a representative 24 bits/pick. GE/pick tracks NAND/pick at $\approx$0.66$\times$. Counts are naive combinational (2-input basis NOT/AND/OR/XOR = 1/2/3/4 NAND; GE 0.67/1.33/1.33/2.67); ARX additions are counted as ripple full adders, rotations as free wiring.

primitive (mode)	out/call	NAND/bit	GE/bit	picks/call	NAND/pick
AES-128-CTR	128 b	810	540	5.3	$2^{14.2}$
ASCON (p^6 keystream)	64 b	600	400	2.7	$2^{13.8}$
SHA-3 / SHAKE128	1344 b	433	289	56	$2^{13.3}$
Salsa20/20	512 b	395	249	21	$2^{13.2}$
Salsa20/8	512 b	167	105	21	$2^{12.0}$

Relative to one full call per pick, squeezing saves a factor equal to picks-per-call: about 2.4 bits for AES, 1.4 for the low-rate ASCON keystream, 4.4 for Salsa, and 5.8 for the wide-rate SHAKE128 (whose 1344-bit rate was almost entirely wasted in the per-call accounting). Because a selection PRG needs only statistical randomness, not cryptographic strength, a reduced-round ARX generator of this class is a legitimate choice, so $\sim 2^{12}$ gates/pick (Salsa20/8) is the realistic figure, not the $2^{16.7}$ of a full AES block.

What squeezing saves in the overall attack: very little. Amortized over the $w \approx 2^{11.2}$ – $2^{14.6}$ monomials of a selected row (wedge and intersection respectively), even the unsqueezed "once per picked row" PRF contributes only $2^{16.7}/w \approx 2^{2.1}$ – $2^{5.5}$ gate-equivalents per monomial; squeezing drops this to $2^{12}/w \approx 2^{-2.6}$ – $2^{0.8}$. Set against address generation at 2^{12} – 2^{14} gates per monomial, the PRF is 2^{12} to 2^{16} times smaller, and it changes the per-monomial total by under 10^{-3} bit. Squeezing harder is a genuine and worthwhile optimization — but of a subterm that was never binding. We therefore set the seed term to zero in all tables below, granting the attacker both the block-structured generator and the squeeze, and note only that neither has been demonstrated on working instances.

The verdict on the memory-free backstop therefore depends on which RNG concession is granted, and two independent optimizations are in play. Squeezing (one wide keystream block carved into many picks, and a reduced-round ARX generator in place of AES) moves the primitive cost from 2^{16} to about 2^{12} per pick, worth 1.8 to 4.0 bits. Block-structured selection — one PRF per *picked row* rather than per row per firing cluster — removes the factor $(R/T)(1+g)$, worth

$2^{7.03}$ at wedge uov-Ip, roughly twice the best possible squeeze. The fan-out removal, not the primitive choice, is the decisive lever:

RNG treatment	RNG/multiply	wedge uov-Ip backstop	vs 143
block-structured and squeezed	$2^{0.76}$	142.6	−0.4
squeezed Salsa20/8, still per-cluster	$2^{7.80}$	143.2	+0.2
squeezed AES, per-cluster	$2^{10.00}$	144.2	+1.2
unsqueezed AES, per-cluster	$2^{11.80}$	145.6	+2.6

So uov-Ip clears the memory-free backstop at every level except the one where the attacker is granted the block-structured generator, and clears it even with the cheapest reduced-round PRG provided the fan-out is still paid. The tables above grant the block-structured scheme and so report 142.6; we record the alternative rather than rely on it, since the memory term carries that cell at +14.4 either way.

4 Assembling the wedge-attack cost with the memory term

4.1 Putting it together, and why the partition is not uniform

Ran's wedge attack [ePrint 2025/1143, §6.2] reduces key recovery to finding a kernel vector of the pruned Macaulay matrix of the projected (v, o', m)-instance, of dimension $N = \binom{v+o'}{v}$ with $\rho = \binom{v+2}{2}$ nonzeros per row, solved by sparse (Wiedemann) linear algebra at a cost he gives as $3 \cdot \rho \cdot N^2 \cdot (2\log_{2^2} q + \log_2 q)$. With his admissible $o' = 18/22/26/32$ this is $N = 2^{60.4}/2^{78.5}/2^{92.8}/2^{117.8}$, reproducing his headline $2^{141}/2^{176}/2^{207}/2^{258}$ — i.e. reductions that sit 2.3 below, 33 above, 0.1 below, and 14.2 below the NIST gate thresholds 143/143/207/272. The decisive point is his stated cost model: *"we assume free access to the coefficients of M."* That is the RAM model; memory is priced at zero. It is exactly the assumption that Baena et al. argue can be *achieved* through their hashing strategy, and the one we have been costing.

Restore the memory term. Each of the $\sim 3 \cdot \rho \cdot N^2$ multiplies reads the length-N vector once; under the [16,40] model that access costs $\rho_m \cdot \psi_1$ with $\psi_1 \approx C_2\big(\log_2 q + \log_2(2N \log_2 q)\big) \cdot \sqrt{N_{\text{part}}}$, $C_2 = 2^{-5}$, and memory reuse factor $\rho_m = 1/n'_r$. By Section 2.1 we take $\rho_m = 1/n'_r$ **with $n'_r =$ 16 to 88**. This is the attacker-favorable choice and it costs us 4 to 6.4 bits; we adopt it because the identity of Section 2.1 leaves no room for the alternative.

That the partition's non-uniformity matters is not a footnote, and it can be computed exactly rather than estimated. The hashing strategy's benefit is that a bucket is smaller than the whole vector: cutting memory into 2^h pieces *of equal size* would replace $\sqrt{N}$ by $2^{-h/2}\sqrt{N}$, a clean $h/2$-bit saving. But the buckets are not equal. Bucket $S \subseteq [h]$ holds the D-subsets meeting $[h]$ in exactly S, so

$$V_S = \binom{N_{\text{gs}} - h}{D - |S|}, \quad \text{with} \quad \sum_S V_S = \binom{N_{\text{gs}}}{D} = N \quad \text{(Vandermonde)}.$$

An access inside S costs proportionally to $\sqrt{V_S}$, and S serves V_S of the N element-accesses, so the access-weighted mean is $(\Sigma_S V_S^{3/2})/N$ against a uniform ideal of $\sqrt{N/2^h}$. Their ratio is the penalty:

$$\text{penalty} \;=\; 2^{h/2}\cdot\sum_{k=0}^{h}\binom{h}{k}\binom{N_{\text{gs}}-h}{D-k}^{3/2}\Big/\binom{N_{\text{gs}}}{D}^{3/2}.$$

This is Baena et al.'s Eq. (15) with the log factors stripped, which cancel in the ratio. By Jensen it is at least 1 with equality iff the partition is uniform, and it is exactly 1 at $h = 0$. Evaluated at $h = 11$:

set	mean bucket overlap	penalty (bits)
uov-Ip	2.30	1.871
uov-Is	2.05	2.190
uov-III	2.07	2.170
uov-V	1.96	2.327

The penalty is linear in h, with a closed form. Sweeping h in unit steps shows penalty(h) is not merely increasing but very nearly proportional to h: a straight line fitted at $h = 11$ tracks it to within 1.0% over h in [8,14] and 3.1% out to $h = 20$. (The true penalty is slightly concave, so the linear law over-states it at large h and therefore *under*-states the attacker's saving; all tables use the exact sum, never the fit.) The reason is that each hash bit acts independently: a column's index set contains any given coordinate with probability $p = D/N_{\text{gs}}$, so the bucket-size distribution is a product over the h bits and the 3/2-moment factorizes. Writing $e = 3/2$, the penalty per hash bit is

$$\delta(p) = (e-1) + \log_2\!\big(p^e + (1-p)^e\big) = \tfrac{1}{2} + \log_2\!\big(p^{1.5} + (1-p)^{1.5}\big), \qquad \text{penalty}(h) \approx h\cdot\delta(p).$$

This predicts the measured slope to within 0.01 bit: $\delta = 0.176/0.204/0.201/0.215$ against measured 0.164/0.194/0.193/0.208 for Ip/Is/III/V, the small excess arising because the hypergeometric samples without replacement, so real buckets are slightly more balanced than independent Bernoulli bits. Two properties are worth naming: $\delta(1/2) = 0$, so a hash that splits every column evenly costs nothing, and $\delta \to 1/2$ as $p \to 0$ or 1. These attacks sit at $p \approx 0.18$–0.21, well away from the balanced point, which is exactly why the penalty is neither zero nor negligible.

So the realized saving is 0.9 to 2.3 bits short of $h/2$ at $h = 11$. The cut is not perfect, and the imperfection runs against the attacker.

Carrying everything through at $h = 11$ with $\rho_m = 1/n'_r$ gives gate-equivalent security of $2^{157.4}$ (uov-Ip), $2^{203.7}$ (uov-Is), $2^{240.3}$ (uov-III), and $2^{304.0}$ (uov-V) — +14.4, +60.7, +33.3, and +32.0 bits above their thresholds. The RAM count places uov-Ip and uov-III essentially on their thresholds, and it is precisely the memory term Ran sets to zero — and Baena et al. try to erase — that is holding the line.

On the $M^{1/3}$ escape hatch. It is tempting to replace the $\sqrt{M}$ of the two-dimensional model by $M^{1/3}$ and reclaim a factor — the memory "is really three-dimensional." But a solid three-dimensional block of active storage is not a thing one can build at this scale, and the obstruction is not engineering timidity, it is two conservation laws pointing the same way. Power must be delivered *in* and waste heat conducted *out*, and both cross the *surface*: a cube holding M bits has volume $\propto M$ but bounding area $\propto M^{2/3}$, so a fixed power density demands $\propto M$ watts in and out while the surface can pass only $\propto M^{2/3}$; the deficit is $M^{1/3}$, and for a uniformly

dissipating solid the interior temperature rise grows like the square of the linear size, $\propto M^{2/3}$, so the core simply cooks. At the sizes this attack needs ($M = N \log_2 q$ with $\sqrt{N} = 2^{30}$ – 2^{59}) that missing $M^{1/3}$ is a factor of 2^{20} to 2^{39} in power and cooling, and no interconnect trick — fiber, immersion, microfluidics — closes a thermodynamic gap of a factor 2^{20}. This is exactly why every physically realized "3D" technology (die stacking, warehouse-scale machines) is in truth quasi-planar: thin active sheets with power and coolant interleaved, whose asymptotics revert to $\sqrt{M}$, or worse once the cooling volume is charged for. So the honest exponent at cryptographic scale is 1/2, and $M^{1/3}$ is a convenient fiction, not a cost model. *Even granting it*, though, the conclusion survives: repricing with $N_{part}^{1/3}$ in place of $\sqrt{N_{part}}$ (and the non-uniform weighting $\Sigma_k \binom{h}{k} V_k^{4/3}$ in place of $V_k^{3/2}$) gives $2^{148.5}$ (uov-Ip), $2^{191.7}$ (uov-Is), $2^{225.9}$ (uov-III), and $2^{285.3}$ (uov-V) — still +5.5, +48.7, +18.9, and +13.3 bits *above* the thresholds. The three-dimensional model is the most generous assumption the attacker can ask for, it is not physically available, and it still leaves every UOV set above threshold once memory is priced at all.

4.2 Check-work: uov-Is, the set the wedge attack does not reach

For completeness we run the same accounting on uov-Is $(q, v, o, m) = (16,96,64,64)$, the one affected set over F_{2^4} rather than $\mathbb{F}_{2^8}$. Here Ran's admissible $o' = 22$ gives $N = \binom{118}{96} = 2^{78.5}$, $\rho = \binom{98}{2} = 2^{12.2}$, and a cheaper field multiply $2(\log_2 q)^2 + \log_2 q = 2 \cdot 4^2 + 4 = 36 \approx 2^{5.17}$, for a RAM cost $2^{175.9}$ — reproducing Ran's 176. The point worth stating plainly: this is *already* +33 bits above the level-I threshold 143, because on uov-Is the wedge attack is not the binding one — the previous-best (intersection/direct) attack at 2^{143} is what pins the set, and $2^{176} > 2^{143}$ means the wedge attack simply does not reduce its security. So uov-Is needs no memory rescue at all. Pricing memory anyway only widens the gap: at $h = 11$, $\rho_m = 1/n_r'$ the total is $2^{203.7}$ in the 2-D model and $2^{191.7}$ even in the physically unavailable 3-D model. Every figure sits far above 143.

5 The Furue–Ikematsu intersection attack, and the all-in cost

Furue and Ikematsu [ePrint 2026/298] push harder than Ran. Their intersection attack, phrased over the p^ℓ − *truncated* ring $\mathbb{F}_q[x]/\langle x_1^{p^\ell}, \dots \rangle$, claims RAM-model gate counts of 128/159/182/223 for uov-Ip/Is/III/V — below Ran's wedge figures and, for three of the four sets, below the NIST thresholds by 10–50 bits. Whether the attack is sound is debatable; here we grant it and ask the only question we have been asking throughout: is that RAM count the whole cost?

The claim is arithmetically self-consistent, and it is a squarefree XL. For every UOV parameter their optimal truncation is $\ell = 1$ over $p = 2$, i.e. $p^\ell = 2$: the ring is $\mathbb{F}_q[x]/\langle x_i^2 \rangle$, a *straight XL with no repeated variables*. In that ring the degree-d monomial count collapses to $\binom{N}{d}$, so their square Wiedemann system has dimension $N_{col} = \binom{n-k}{D} - \binom{2o-v-k}{D} + 1$ and cost $3 \cdot \binom{n-k+1}{2} \cdot N_{col^2} \cdot (2 \log_{2^2 q} + \log_2 q)$. Plugging in their optimal (ℓ, k, D) reproduces the table to a fraction of a bit: uov-Ip (1,7,13) gives $N_{col} = 2^{53.6}$ and $2^{128.4}$; uov-Is (1,16,16) $\to 2^{158.5}$; uov-III (1,14,18) $\to 2^{181.5}$; uov-V (1,22,22) $\to 2^{223.3}$. So the 2^{128} is real *as a field-operation count* — and, exactly as in Ran's estimate and Baena et al.'s, memory is priced at zero. Because the columns are size-D subsets of [n-k], every cost we have developed applies verbatim.

Two overheads, and which one dominates. Per monomial-touch the field multiply is $2^{7.09}$

gates; address generation after the $1/n'_r$ reuse is $2^{6.4}$ – $2^{7.1}$ gate-equivalents; and one partitioned memory access is 2^{20} – 2^{43} gate-equivalents. So the on-the-fly cost refutes the "negligible" claim relative to the field multiply — it is 9 to 11 bits above the asserted sub-0.1% on these sets — but it is decisively *dominated by the memory term.* The consequence is that the all-in cost exceeds the memory-only cost by well under a bit. This matters for honesty in both directions. Against "on-the-fly is free," the address term alone is a genuine backstop worth 0.7 to 1.9 bits on the RAM count for these sets, even if the memory model were rejected outright. But against over-claiming, that backstop is now far too small to carry a verdict: the security margins below rest on the memory term.

The numbers, and the sets that are not safe. At $h = 11$, $\rho_m = 1/n'_r$. The "on-fly, mem off" column discounts memory entirely († marks below-threshold):

set	N_{col}	RAM (Furue)	on-fly, mem off	+mem (2-D)	+mem (3-D)
uov-Ip	$2^{53.6}$	$2^{128.4}$†	$2^{129.1}$†	$2^{141.3}$†	$2^{133.1}$†
uov-Is	$2^{69.2}$	$2^{158.5}$	$2^{160.4}$	$2^{181.1}$	$2^{170.2}$
uov-III	$2^{79.5}$	$2^{181.5}$†	$2^{182.4}$†	$2^{207.5}$	$2^{194.8}$†
uov-V	$2^{100.0}$	$2^{223.3}$†	$2^{224.2}$†	$2^{259.6}$†	$2^{243.4}$†

Plainly: under this attack uov-Ip and uov-V are not safe on our accounting, and uov-III clears by only half a bit. The margins are thin because of the reuse factor of Section 2.1, which we have granted the attacker. Granting the attack itself, the physical (2-D) memory model at $\rho_m = 1/n'_r$ leaves uov-Is comfortably clear and uov-III marginally clear, with uov-Ip short by 1.7 bits and uov-V by 12.4. The defense of those two sets therefore does not come from this note's cost additions; it must come from one of three places, which we name rather than obscure.

(i) *The attack's soundness is unverified.* Its complexity rests on the solving degree D predicted by Furue's semi-regularity Conjecture 1, and his own §4.4 already exhibits a UOV instance where the predicted rank misses the experiment by a degree; no mid-range *working* example of the full attack at these parameters has been published. Since $N_{col} = \binom{n-k}{D}$, a true solving degree one higher than predicted multiplies N_{col} by $(n-k-D)/(D+1) \approx 7$ — about +2.8 bits, or +5.6 on the squared cost — so a single-degree error in the conjecture more than erases uov-Ip's deficit and covers about half of uov-V's. A complexity that drives a parameter-security verdict should be demonstrated on solvable instances before it is treated as a lower bound; that standard has not been met here. Section 5.1 sharpens this: at the optimum there is no next degree to slip to.

(ii) *Our memory constants are not the tightest against the attacker.* We take $\rho_m = 1/n'_r$ at its full value and $C_2 = 2^{-5}$ unmodified, and we do not reach for $\rho_m > 1$ although a coordinate touched by many nonzeros per mat-vec arguably invites it.

(iii) *A bit or two is inside the modeling noise* of estimates built on order-of-magnitude constants — though 12.4 bits, for uov-V, is not.

We do not pretend the memory argument alone settles uov-Ip or uov-V under this attack. It does settle every set under Ran's wedge attack, which is the more widely accepted of the two.

Why these structural attacks are, in a real sense, weaker than plain FXL: they cannot be shrunk to fit the memory you have. There is a deeper reason to doubt that either the Furue intersection attack or the Ran wedge attack is the one to worry about, and it is about *reachable* memory. A generic MQ solver has a dial the structural attacks lack. In FXL —

fix-then-XL, the guess-and-solve method sometimes misattributed as "hybrid F5" — one guesses g of the variables and runs XL on the remaining n-g; guessing more shrinks the Macaulay matrix (hence the memory) as far down as one likes, at a time cost of q^g per guess. The attacker with a fixed memory budget M simply raises g until the system fits in M, and pays for it in time. FXL therefore *always* presents a memory goal it can meet; that is the honest way to state a cost against a real machine. The Furue and Ran attacks have no such dial. They are structural: they work *because* the system is a UOV public key, exploiting the oil-space intersection (Furue) or the exterior-algebra annihilator (Ran), and their matrix dimension is fixed by that structure and the solving degree. The one guess parameter they do have is bounded by the structural slack — $k < 2o - v$ for Furue, the projection $o' < o$ for Ran — so it cannot reduce n-k below n-(2o-v), and the dimension $\binom{n-k}{D}$ (or $\binom{v+o'}{v}$) has a hard *floor*. Guess past the slack and the structure the attack relies on is gone; you are back to generic FXL, slower. So the memory these attacks demand is not negotiable.

What is that floor, as physical memory? The Wiedemann stage must hold a length-N vector over $\mathbb{F}_q$, i.e. $N \cdot \log_2 q$ bits at minimum (the whole point of building the matrix on the fly is to avoid storing more). Taking 2^{44} bits — a 2 TB big-memory server — as the "easily acquired" unit:

set	Furue memory (bits)	Furue servers	Ran memory (bits)	Ran servers
uov-Ip	$2^{56.6}$ (13 PB)	$2^{12.6}$	$2^{63.4}$ (1.4 EB)	$2^{19.4}$
uov-Is	$2^{71.2}$	$2^{27.2}$	$2^{80.5}$	$2^{36.5}$
uov-III	$2^{82.5}$	$2^{38.5}$	$2^{95.8}$	$2^{51.8}$
uov-V	2^{103}	2^{59}	$2^{120.8}$	$2^{76.8}$

The level-I demands are large but *acquirable*: $\sim 2^{13}$ – 2^{19} big-memory servers, a data-center to hyperscaler-fleet scale, and 2 TB machines are commodity. The level-V demands are not of this world. Furue's uov-V needs 2^{59} such servers and Ran's needs $2^{76.8}$; there are on the order of 2^{33} computers on the planet, and all the DRAM ever manufactured is perhaps 2^{73} – 2^{77} bits — less than a single uov-V vector for the wedge attack. No aggregation reaches 2^{100} bits, let alone the $\sqrt{M}$ − *priced* random access to it. And because, unlike FXL, these attacks cannot be dialed down to a reachable size, a uov-V "break" at 2^{223} or 2^{258} gate-operations is quoting a number for a machine that cannot be built. It is the attacker's job to present a memory goal that can actually be reached; FXL does, and does not break UOV at these levels; the structural attacks that dip below threshold on paper do not, and demand memories that range from a serious data center (level I) to more bits than exist (level V). That gap — between a costed attack and a physically runnable one — is exactly what the RAM model hides by pricing memory at zero.

5.1 Independent reproduction of the optimizer, and where its accepted cells sit

We re-derived their parameter search rather than quoting it. With $P = p^\ell$, the Hilbert series of their Conjecture 1 factors as

$$\frac{(1 + z + \cdots + z^{P-1})^{n-k}}{(1 + z^2 + \cdots + z^{2(P-1)})^M} \;=\; (1 - z^P)^{n-k}(1 - z^2)^M \cdot (1 - z)^{-(n-k)}(1 - z^{2P})^{-M},$$

so that no length-P convolution is ever formed and the whole search runs in exact integer arithmetic. It reproduces their Table 1 — the reconciliation grid on $(q, v, o, m) = (16,40,20,30)$ — in every filled cell and every blank, and all eight cells of their Table 3, 141/176/208/258 for reconciliation and 128/159/182/223 for the intersection, at their exact optimal (ℓ, k, D). We therefore take their arithmetic as given; what follows is about *where* the optimum sits, not whether it was computed correctly. (One cell of their Table 6 does not reproduce: for their own replacement set uov-Ip' = (256,120,44) the search finds $(\ell, k, D) = (5,11,30)$ at $2^{222.5}$, 1.3 bits under the (2,2,29) = $2^{223.8}$ they report. Too small to matter, and it does not touch Table 3.)

Every accepted cell sits on the last admissible degree. The framework admits $2 \le D \le \alpha(p^\ell - 1)$, with $\alpha = 2o - v - k$ the intersection dimension; at the optimal $\ell = 1$ that window closes at $D = \alpha$. In all four Table 3 cells the chosen D is exactly α — the last degree the framework allows — and it is accepted at the first *non-positive* coefficient of the series, one step after a coefficient still 40 to 90 bits above the target $T(\alpha, 2, D-1) = \alpha$:

set	(ℓ, k, D)	$\alpha = 2o - v - k$	$[G]_{D-1}$	target α	$[G]_D$
uov-Ip	(1,7,13)	13	$+2^{44.8}$	13	$-2^{47.7}$
uov-Is	(1,16,16)	16	$+2^{58.2}$	16	$-2^{62.4}$
uov-III	(1,14,18)	18	$+2^{69.9}$	18	$-2^{69.2}$
uov-V	(1,22,22)	22	$+2^{89.0}$	22	$-2^{87.5}$

This is not by itself an error: a coefficient that has gone negative means only that, under semi-regularity, the ideal is predicted to fill the whole degree-D component, so that whatever kernel survives is the solution-derived part the attack wants, and a steeply overdetermined series (here $M = 3m - 2$ equations against n variables) does drop abruptly. We scanned all 270 feasible cells over $\ell \le 3$ for the four sets and *every one* of them is accepted the same way; the behavior is systematic, not a quirk of the optimum. But it does mean the solving degree is never an interior estimate. There is no degree at which the predicted dimension descends *gradually* to the target; the verdict is decided by a single step of a heuristic series, at the one degree where that series can no longer be checked against the framework's own window — and their §4.4 exhibits exactly such a boundary miss, an experimental rank of 918 against a predicted 921.

What a one-degree slip costs, restated. Section 5(i) priced it as +5.6 bits on the squared cost, which is the right figure when a lower-k cell is available to absorb the extra degree. The sharper statement is that at the boundary a further degree does not exist: with $D = \alpha(p-1)$ already, the same k is simply outside the window, and the attacker must move. If the $\ell = 1$ column fails wholesale, the best surviving cells are $2^{165.7}$ (5,19,20), $2^{209.8}$ (4,30,25), $2^{249.9}$ (5,31,30) and $2^{311.6}$ (6,43,37) for Ip/Is/III/V — +37 to +88 bits, which puts all four sets above their NIST thresholds *on the attacker's own RAM accounting*, with no memory term needed. We do not claim that outcome; we record that the distance between it and the published table is one degree of an unverified conjecture.

5.2 Proposed replacement parameters: uov-Ip#, uov-III# and uov-V#

We decline Furue's replacement parameters and propose three of our own, holding m and $|cpk|$ fixed and spending only on the vinegar count v:

	(q, n, m)	$\alpha = 3m - n$	replaces	NIST req.
uov-Ip#	(256,116,44)	16	uov-Ip (256,112,44)	143
uov-III#	(256,186,72)	30	uov-III (256,184,72)	207
uov-V#	(256,250,96)	38	uov-V (256,244,96)	272

The quantity that governs the intersection attack is the intersection dimension itself, $\alpha = 2o - v = 3m - n$: one unit of m buys the attacker *three* dimensions, one unit of n takes back one. Since m fixes $|cpk|$ and the forgery margin, and n fixes α, the two levers are not interchangeable and only the second is cheap. Each of the three sets is the first n at which the framework's solving degree D steps up — 13→14, 18→19, 24→25 respectively — which is where the cost per added variable is best; between steps the figures creep by only ∼0.8 bit per variable.

The costing. All figures are $\log_2$ gate-equivalents at $h = 11$, $\rho_m = 1/n'_r$; (ℓ, k, D) are re-derived, not quoted (Section 5.1); † marks a figure below the NIST requirement; **bold** marks the binding cell.

set	attack	(ℓ, k, D)	N_{col}	RAM	FLY	MEM (2-D)	MEM (3-D)
uov-Ip#	intersection	(1,2,14)	$2^{58.1}$	137.6†	138.4†	152.9	143.9
	wedge (Ran)	$o' = 19$	$2^{64.0}$	148.0	150.1	166.6	157.0
	direct (FXL)	—	—	145.6	147.1	**151.1**	—
uov-III#	intersection	(1,11,19)	$2^{83.4}$	189.3†	190.2†	**217.2**	203.9†
	wedge (Ran)	$o' = 27$	$2^{95.8}$	213.0	215.6	247.9	233.0
	direct (FXL)	—	—	217.9	220.9	241.5	—
uov-V#	intersection	(1,13,25)	$2^{111.6}$	246.7†	248.0†	**289.1**	271.1†
	wedge (Ran)	$o' = 34$	$2^{124.5}$	271.2†	274.3	320.8	301.0
	direct (FXL)	—	—	277.9	281.6	314.1	—

The binding attack is not the same one at every level. At levels III and V it is the intersection, clearing by +10.2 and +17.1 bits over the NIST requirement; blocking (Section 6) costs the defender 0.8 and 1.0 of those, leaving +9.4 and +16.1. **At level I it is the forgery**: uov-Ip#'s memory-charged direct attack sits at $2^{151.1}$, 1.8 bits *below* its intersection at $2^{152.9}$, so the margin there is +8.1 and not +9.9. Blocking does not change that verdict — it moves the intersection to +9.2, still the weaker constraint — and the reversal is the intended consequence of holding m fixed while spending on v: pushed far enough, the structural attacks stop being what binds. Against the sets they replace the gain is +11.6, +9.7 and +29.5 bits of MEM, which is what turns three verdicts of −1.7, +0.5 and −12.4 into three that are not decided by sub-bit effects. uov-Ip# additionally clears in the physically unavailable 3-D model, by 0.9; uov-III# and uov-V# do not, by 3.1 and 0.9, and $n = 188$ or $n = 252$ would buy that column at +2.2% and +1.6% more key if it is wanted.

What still leans on this note, stated plainly. All three clear on the memory term and none clears on RAM: the intersection deficits are 5.4, 17.7 and 25.3 bits, and Section 2's address-generation term closes none of them. uov-V#'s wedge cell is 0.8 bit under its requirement on RAM and clears on FLY at 274.3, i.e. it needs the matrix-regeneration argument but not the memory-access model. The wedge cells at levels I and III, by contrast, clear on RAM alone, by 5.0 and 6.0 bits. These are the cells a reader who rejects Section 4 will attack, and we would rather name them than have them found.

Degree robustness. Each optimum again sits at $D = \alpha(p-1)$ with $\alpha - k = D$, accepted at the first non-positive Hilbert coefficient (Section 5.1). A solving degree one higher than conjectured

moves the attacker up the k-ladder rather than off it: to $2^{143.5}$, $2^{195.6}$, $2^{253.1}$ respectively, and two degrees to $2^{149.2}$, $2^{201.8}$, $2^{259.4}$. So the RAM figures are soft by 5.9–6.4 bits per degree *in the defender's favor*, with 3, 12 and 14 admissible $\ell = 1$ cells of fallback before the column closes and the cost jumps to $2^{194.0}$, $2^{262.4}$, $2^{349.2}$.

The machines. The Wiedemann vectors are $2^{61.1}$, $2^{86.4}$ and $2^{114.6}$ bits: $2^{17.1}$ of the 2 TB servers of Section 5 at level I, $2^{42.4}$ at level III, $2^{70.6}$ at level V — the first a large data center, the second some thousands of times all the DRAM ever manufactured, the third beyond it by 2^{38} – 2^{42}. The Section 8 latency floors at $k' = 1$ are 281 years, 3.8×10^{12} years and 8.6×10^{23} years.

The price. m is unchanged at every level, so $|cpk|$ — which depends only on q and m — does not move at all:

	$\|epk\|$	$\|esk\|$	$\|cpk\|$	$\|\sigma\|$
uov-Ip (256,112,44)	278,432	237,896	43,576	128
uov-Ip# (256,116,44)	298,584	258,224	43,576	132
uov-III (256,184,72)	1,225,440	1,044,320	189,232	200
uov-III# (256,186,72)	1,252,152	1,071,176	189,232	202
uov-V (256,244,96)	2,869,440	2,436,704	446,992	260
uov-V# (256,250,96)	3,012,000	2,579,840	446,992	266

That is +7.2%, +2.2% and +5.0% on the expanded public key, 4, 2 and 6 bytes on the signature, and nothing on the compressed public key. Furue's own replacements cost +14.7%, +13.4% and +13.5% on $|epk|$ and 8, 12 and 16 bytes of signature to reach the same verdicts without the memory argument.

The forgery side, and why m is held fixed. Raising m instead of n would be the natural instinct and is the wrong move: at fixed n it *shrinks* v, and every key-recovery attack on UOV is governed by the vinegar count. For (256,112,45) the intersection falls to $2^{121.2}$ RAM and $2^{132.3}$ MEM — 7.2 and 9.0 bits *worse* than uov-Ip — against a forgery gain of only 1.8 bits. Holding m does leave the forgery margin where it was, and we have checked that this is enough. The direct attack is costed by the round-2 specification's own §4.2 model (Thomae–Wolf to m-1 equations in m-1 variables, then hybrid Wiedemann XL), which reproduces its Table 5 — 145/165/218/278 — and by Hashimoto's algorithm in the form of the SNOVA round-2 specification §5.1, which reproduces all nine of its Table 8 cells. Hashimoto's transformation needs $n \geq (\alpha_H + 1)(m - k - \alpha_H + 1)$, so the underdeterminedness is its fuel and it first beats Thomae–Wolf at $n = 117$, 192 and 264 for $m = 44$, 72 and 96. **All three sets sit below those thresholds** — uov-Ip# by a single variable — so the direct figures in the table above are flat in n, at 145.6/217.9/277.9, above requirement by 2.6, 10.9 and 5.9 bits with memory priced at zero and by 8.1, 34.5 and 42.1 with it charged. That single variable is not slack we chose to leave: at (256,117,44) the direct attack falls to $2^{142.2}$, below the level-I requirement, so **$n = 116$ is the largest v that $m = 44$ will carry** in the RAM model. It is also, not coincidentally, close to where the intersection stops being the binding attack — at $n = 117$ the $\ell = 1$ column has one cell left and the intersection reaches $2^{144.3}$ on RAM alone. The two attack families cross within a variable of each other, and uov-Ip# is placed on the last n before the forgery side gives way. Note the asymmetry that makes this the right place to stop: unlike the structural attacks, FXL can always shrink to the memory it has and pay q^k for it, so charging it memory adds only 5.5 to 36 bits and its vectors stay at $2^{42.8}$ – $2^{97.0}$ bits. Its number is a number for a machine that

can be built, which is exactly why it, and not the intersection, should be the constraint that binds.

Ran's wedge attack, and how o' is fixed. Ran's projected dimension o' needs no separate search: it falls out of the one already run. In the p^ℓ-truncated framework at $\ell = 1$ over $p = 2$ the reduced symmetric algebra *is* the exterior algebra, so Ran's attack is the $\ell = 1$ reconciliation cell with $k = o - o'$ and $D = o'$; his admissible $o' = 18/22/26/32$ are exactly the reconciliation optima our own search returns for the four round-2 sets, in each case with $o - k = D$, the boundary cell of Section 5.1 once again. Two accounting differences remain: Ran prunes the column $\binom{o'}{o'}$, giving $N = \binom{v+o'}{o'}$ exactly, and he charges the tighter row weight $\binom{v+2}{2}$ rather than $\binom{n-k+1}{2}$, because the alternating polar form couples only through the vinegar-indexed part (Section 2.5). Both are in his favor and both are granted. Running our search on the three new sets returns $o' = 19$, 27 and 34, and the wedge figures in the table above.

The wedge is now the binding structural attack on all three sets, and it clears. At $2^{148.0}$, $2^{213.0}$ and $2^{271.2}$ it sits above the intersection by 10.4, 23.7 and 24.5 bits on RAM — as it does on the round-2 sets — so the intersection remains the number to beat, but the wedge is what pins the *RAM*-model verdict, and at levels I and III it clears with memory priced at zero (+5.0 and +6.0). Only uov-V# fails there, by 0.8, and Section 2's address term closes that at $2^{274.3}$. The wedge machines are larger than the intersection's throughout: $2^{67.0}$, $2^{98.8}$ and $2^{127.5}$ bits against $2^{61.1}$, $2^{86.4}$ and $2^{114.6}$, which is to say that even the attack whose gate count is more favorable to the defender is the one demanding the more impossible memory.

What remains uncosted. The MinRank and Kipnis–Shamir attacks, which the round-2 specification places 60 to 170 bits above the binding attack on every set and which move in the defender's direction under an increase in v; and the collision attack, which depends only on q and m and is therefore unchanged at 191/303/399. We do not restate them here.

5.3 The alternative if the memory argument is rejected

One further set, $(q, n, m) = (256,128,48)$, is worth costing, because the reason we do not propose it is the reason Section 5.2 holds m fixed. (256,128,48) has $\alpha = 16$ and gives reconciliation (1,27,21) at $2^{163.3}$ and intersection (1,0,16) at $2^{154.4}$, the latter binding: it clears level I by 11.4 bits *with memory priced at zero*, and MEM lifts it to $2^{173.8}$ and the 3-D model to $2^{163.4}$. It is the strongest set in this note by some margin, and it is the only one that needs no part of this note's accounting. But it buys that by raising m from 44 to 48, which costs +42.3% on $|epk|$ and, uniquely among the sets here, +29.6% on $|cpk|$ — 56,464 bytes against 43,576 — since $|cpk|$ scales as m^3. uov-Ip# reaches a defensible verdict for +7.2% and no change to $|cpk|$ at all. The comparison is worth keeping in view if the memory argument of Section 4 is rejected: (256,128,48) is then the fallback, at roughly six times the key-size cost.

It is also an instructive degenerate case for Section 5.1. With $\alpha = 16$ and solving degree 16, $k = 0$ is the *only* admissible cell at $\ell = 1$ — no slack to spend on guessing and, by that section, no room for a further degree — so a single slip in Conjecture 1 puts the attacker on (5,15,28) at $2^{216.3}$, a 62-bit cliff. The three sets of Section 5.2 sit further up the ladder, with 3, 12 and 14 cells of fallback, and are correspondingly less exposed to a conjecture we do not believe has been established.

6 Block Wiedemann

Coppersmith's block Wiedemann advances k' vectors together. It changes no arithmetic: $(N/m+N/n)$ sequence steps plus N/n reconstruction steps, each applying M to n vectors, is 3N matrix-vector equivalents for any k', identical to classic Wiedemann, so $3\rho N^2$ stands. Its usual selling point — parallelism across machines with no communication during the Krylov phase — is worth nothing in a model that charges resource·time rather than wall-clock.

What it does buy, per matrix entry, is locality. One address is computed and used for k' multiply-accumulates, so **FLY is divided by** k'. And one access fetches the k' block-components of a coordinate as $k' \cdot \log_2 q$ *consecutive* bits, so the address field $\log_2(memory)$ — which is 58 to 122 bits against a payload of only $\log_2 q = 4$ or 8 — is paid once per k' multiplies instead of once each. What it costs is that the working set is k' times larger, so $\sqrt{\text{memory}}$ grows as $\sqrt{k'}$. Per useful multiply the memory term is therefore proportional to

$$f(k') = \left(\log_2 q + A/k'\right) \cdot \sqrt{k'}, \qquad A = \log_2(2\,k'\,N \log_2 q),$$

minimized at $k'^* = A/c$ with $c = \log_2 q$, where $f(1)/f(k'^*) = (c + A)/(2\sqrt{Ac})$. That ratio is exactly the **arithmetic-geometric mean gap** between the payload width c and the address width A: blocking always helps weakly, buys nothing when $c = A$, and buys more the more lopsided they are — but only as $\sqrt{A/c}$, which is why 58 to 122 bits of address overhead yields under 1.3 bits.

	$\log_2 q$	A	k'^*	MEM saving	FLY saving	footprint cost
wedge uov-Ip	8	64.4	8	0.64	3.00	+3.00
wedge uov-Is	4	81.5	20	1.21	4.32	+4.32
wedge uov-III	8	96.8	12	0.89	3.58	+3.58
wedge uov-V	8	121.8	15	1.03	3.91	+3.91
int. uov-Ip	8	57.6	7	0.58	2.81	+2.81
int. uov-Is	4	72.2	18	1.12	4.17	+4.17
int. uov-III	8	83.5	10	0.79	3.32	+3.32
int. uov-V	8	104.0	13	0.93	3.70	+3.70

RNG is **not** amortized by k': row selection is a per-(row, cluster) decision and does not care how many vectors are advanced. So blocking reduces FLY and MEM but leaves the seed term untouched.

At the optimal k' the wedge totals become 156.8/202.5/239.5/303.0 — margins +13.8/+59.5/+32.5/+31.0, unchanged in substance. The intersection totals become 140.7/180.0/206.7/258.6. **The one cell that changes sign is the intersection attack on uov-III, from +0.5 to −0.3.** That cell was never robust; a sub-bit effect either way decides it.

Against this, blocking makes Section 5 strictly worse for the attacker: the working set grows by k', so the physical footprint rises 2.8 to 4.3 bits — wedge uov-Ip from $2^{63.4}$ to $2^{66.4}$ bits ($2^{19.4}$ to $2^{22.4}$ servers), uov-V from $2^{120.8}$ to $2^{124.8}$. The attacker buys about one bit of gate cost with three to four bits of memory, which is the resource Section 5 argues cannot be obtained.

7 Why h cannot simply be raised: the units stop being closely coupled

Every figure above is at $h = 11$: a partition into $2^{11} \approx 2000$ processing units, few enough to sit in one box, wired together directly, packed in local layers so the hops are short. The natural objection is that the attacker should just raise h — a larger partition cuts the memory term's $2^{-h/2}$ factor, so surely the MEM column falls.

It does, and we should be precise about how. By the closed form of Section 4 the net memory saving is

$$S(h) = h/2 - \text{penalty}(h) \;\approx\; h \cdot \big(\tfrac{1}{2} - \delta(p)\big),$$

linear in h, with no diminishing returns whatever. The per-bit rate 1/2 - $\delta(p)$ is 0.336/0.306/ 0.307/0.292 bits for Ip/Is/III/V, so $S(11) = 3.6/3.3/3.3/3.2$ bits and every further hash bit buys another ≈0.3. (Those are the wedge rates; the intersection sets sit at smaller $p = D/N_{\text{gs}} \approx 0.10$ – 0.12, where $\delta(p)$ is nearer 1/2, and their exact marginals are 0.18 – 0.22 bits per hash bit — smaller, but equally free of an interior optimum.) Worse for us, the marginal rate does not decay — it drifts *up* slightly, from 0.333 at $h = 8$ to 0.366 at $h = 40$ for uov-Ip — because $1/2 - \delta(p) > 0$ for every p strictly inside $(0, 1)$. In the 3-D model the same holds at rate 1/3 - $\delta_{4/3}(p) \approx 0.21$. **There is no interior optimum: the memory model on its own rewards an unboundedly large partition.** Two things outside it supply the cap.

The first is hardware. The $2^{-h/2}$ saving is a saving *only if the 2^h units are closely coupled* — if any unit can reach the accumulator in a short, cheap, wire hop. That regime has a hard ceiling, and it is a low one. The densest wire-coupled single-box FPGA cluster we are aware of is the SciEngines RIVYERA S6-LX150, which tops out at 256 Xilinx Spartan-6 LX150 FPGAs on one backplane [SciEngines] — that is 2^8 units, and even there the machine is built from cards of 8 compute FPGAs plus a ninth FPGA per card spent purely on communication, so coupling already costs silicon inside the box. A thousand (2^{10}) is a generous upper bound on what stays directly wired. So $h = 11$ already sits at the very edge of the closely-coupled regime, and $h = 16$ is two orders of magnitude past it: by the vendor's own terminology, anything larger is no longer one machine but a "RIVYERA cluster ... interconnected by network" [SciEngines], i.e. exactly the long-distance transmission regime that Section 9 prices.

The second is the fan-out of Section 2.4, which makes the impossibility concrete. At $h = 16$ each row must combine partial results across $\Theta(h^2) \approx 137$ clusters; once those clusters no longer share a box, that is on the order of a hundred inter-box links *per chip*. As Section 9 argues, a single-mode fiber is ∼125 μm of cladding against a ∼10 nm on-chip wire, and each optical hop carries a transceiver energy floor of order $10^2 - -10^3$ gate-operations per bit. Crowding a hundred fiber cables onto every chip — to buy back the bit or two that raising h promised — is plainly not a solution: it does not fit, it does not power, and its per-hop cost exceeds the field multiply it serves.

The two effects together give the exchange rate the attacker actually faces. S(h) grows linearly while the fan-out g(h) = 1 + (h-a) + $\binom{h-a}{2}$ grows quadratically, so the coordination paid per bit of memory bought rises without limit:

h	S(h) bits	fan-out g	g per bit saved	clusters
8	2.63	22	8.4	2^8
11	3.63	46	12.7	2^{11}
16	5.32	106	19.9	2^{16}
20	6.70	172	25.7	2^{20}
40	13.83	742	53.7	2^{40}

Doubling the partition costs about four times the coordination for about one more bit. That, and not the memory term, is what caps h. The honest ceiling on the closely-coupled partition is therefore $h \approx 10$–11, which is what the tables use. But the conclusion does not depend on that ceiling being tight: taking the memory model at face value and letting the attacker raise h freely, Ran's wedge attack still clears every threshold for all $h \leq 62$ (uov-Ip), $h \leq 95$ (uov-Is), $h \leq 111$ (uov-III), and $h \leq 116$ (uov-V) — and 2^{40} clusters is already about 10^{12}, far beyond any buildable machine. No verdict in this note turns on the choice of h.

8 Latency and distance: a time–memory trade-off like AT^2

The accounting above prices gates and memory accesses. Two physical constraints remain unpriced, and both are independent of every constant in dispute: the time it takes light to cross the machine, and the energy it takes to push a bit through a transceiver.

Thompson's bound constrains **bandwidth**: a bisection of width $\Theta(\sqrt{A})$ must pass B bits, so $A \cdot T^2 \geq B^2$. What follows is its **latency** counterpart. The Wiedemann recurrence $v_{i+1} = M \cdot v_i$ is a strict dependency chain, and each link is a global gather — every output coordinate sums a row support scattered across the whole vector — so no step can complete in less than one light-crossing of the machine:

$$T \;\geq\; D \cdot L/c,$$

with L the machine's linear extent. Everything therefore turns on how large the memory physically is, so we build it from parts one can buy.

The machine. Take the intersection attack on uov-Ip. The vector alone is $2^{53.6}$ bytes, about 14 PB. A fully decked-out air-cooled 1U server carries 4 TiB of DRAM (32 × 128 GiB DIMMs); a 42U rack of them holds 168 TiB — and at roughly a kilowatt per server the rack draws ∼40 kW, already past what plain air moves, so "fully decked out and air-cooled" is itself a concession. The vector needs **76 racks** ($2^{53.63}/2^{47.39}$ bytes; the rounded $2^{53.6}$ of the tables would give 74). At ≈ 1.5 m^3 per rack (0.6 × 1.2 × 2.1 m; switches and power assumed to live elsewhere, free) that is 114 m^3 of hardware, a density of ∼0.12 PB/m^3. Idealize it as a sphere — the shape that minimizes the diameter, hence the most attacker-favorable layout, aisles and plumbing charged nothing — of radius

$$r = \left(\frac{3V}{4\pi}\right)^{1/3} = 3.0 \text{ m},$$

and then grant *extreme* cooling — immersion, microfluidics, whatever it takes — by **halving the radius** to 1.5 m: an 8× volumetric density gain, to ≈ 1 PB/m^3. That is as far as the concession can honestly stretch: bare DIMM silicon packs about 4 PB/m^3 with zero volume left

for power, cooling, or interconnect, so one more halving would exceed the density of the memory chips themselves. (Packing the vector at 14 PB/m^3 into a single cubic meter — denser than bare DIMM silicon itself — shortens the crossing and *weakens* the floor to 4.5 years; the script reports that attacker-favorable variant alongside this one.)

The floor. One light-crossing of the cooled sphere — diameter 3.0 m — takes 10.0 ns in vacuum (15 ns in fiber; we charge vacuum). The chain is $D = 3\mathrm{N} = 2^{55.2}$ strictly dependent steps, so

$$T \;\geq\; 2^{55.2} \times 10.0 \text{ ns} \;\approx\; \textbf{13 years},$$

on a machine assumed to compute instantaneously and with unlimited parallelism. The only cost charged is the propagation of light through a machine the size the memory forces it to be.

In cycles per useful multiplication. At 3 GHz the crossing is ∼30 clock cycles per chain link. The irreducible arithmetic depth of a link is one field multiply plus a $\lceil \log_2 \rho \rceil = 13$-level addition tree — about 14 cycles — so even with a multiplier for every matrix entry, the machine waits two cycles on light for every cycle of forced arithmetic. And merely *meeting* the floor requires sustaining $3\rho N^2 \,/\, (T \cdot f) = 2^{61}$ **multiplies per clock cycle** — some 2.6 exa-multipliers, about 2^{68} gates of multiplier silicon alone, running flat out for 13 years. Fewer multipliers push T above the floor; more buy nothing, because the chain, not the arithmetic, is what binds.

The energy of transmission (TRANS). Latency aside, the partial sums of Section 2.4 must physically move. Each selected row scatters $g = 46$ partials across its fan-out ball; each partial leaves its generating cluster through a transceiver, carrying $\mathrm{B} = \log_2 q + \lceil \log_2 N \rceil$ bits (payload plus row tag), and we charge 2^7 **gate-energies per bit per hop** — the *cheap* end of Section 9.2's 10^2 – 10^3 range for the modulator/laser + photodiode + TIA + SerDes chain, i.e. roadmap silicon photonics; purchasable optics runs 5 – 7 bits higher (Section 9.2). The total over the attack is

$$\mathrm{TRANS} = 3N^2 \cdot g \cdot B \cdot 2^7 \text{ gate-equivalents, i.e. } g\,B\,2^7/w \text{ per useful multiply.}$$

For the intersection on uov-Ip that is $2^{6.0}$ per multiply against a $2^{7.1}$ field multiply; for the wedge on uov-Is it is $2^{6.7}$ against a $2^{5.2}$ multiply — over $\mathbb{F}_{16}$ the transceivers cost *more than all the arithmetic they serve*. Across the eight cells TRANS lands within −2.4 to +1.5 bits of the entire RAM-model gate count (table below): moving the partial sums costs about as much energy as doing the mathematics. So even a reader who rejects the memory-access model of Section 4 outright, and the address-generation accounting of Section 2 with it, still owes a transmission term of the same order as the count being defended. Two caveats, one in each direction. For the attacker: a perfectly systolic schedule could strip the row tag, dropping B to $\log_2 q$ and TRANS by 3.0 to 4.4 bits, and block Wiedemann amortizes the tag by k' — the same arithmetic–geometric gap as Section 6. For the defender: the 2^7 endpoint charge already assumes optics that cannot yet be bought, intra-rack electrical hops are uncharged, and each hop was counted once rather than as TX and RX separately. Added to the memory-free backstop of Section 3, TRANS moves the wedge uov-Ip figure from 142.6 to **143.1** — the backstop crosses the NIST threshold with memory still priced at zero.

Block Wiedemann does not escape the latency floor; it converts it into a trade-off. Blocking width k' cuts the depth to $3N/k'$, but multiplies the memory by k' and so the linear size by $k'^{1/3}$. The two effects do not cancel:

$$T(k') = \frac{3N\,L_0\,k'^{1/3}}{c\,k'} = \frac{T(1)}{k'^{2/3}}, \qquad \text{hence} \qquad \boxed{T \cdot M^{2/3} \;\geq\; \frac{3N\,M_0}{c\,\rho_d^{1/3}}},$$

with M_0 the unblocked vector size and $\rho_d \approx 2^{52.8}$ bits/m^3 the cooled rack density above. This is the same moral as AT^2, reached by the other route: halving the time costs a factor $2^{3/2} \approx 2.83$ in memory, exactly as halving Thompson's T costs a factor 4 in area. Time is purchasable only with geometry, and here the geometry is memory — the resource Section 5 argues cannot be obtained.

	T at k'=1	k' for one year	memory then	vs all DRAM (2^{73} b)	TRANS	vs RAM
int. uov-Ip	13.3 yr	49	$2^{62.2}$	−10.8	$2^{127.3}$	−1.1
int. uov-Is	1.9×10^7 yr	8.4×10^{10}	$2^{107.5}$	+34.5	$2^{158.8}$	+0.2
int. uov-III	3.3×10^{11} yr	1.9×10^{17}	$2^{139.9}$	+66.9	$2^{179.6}$	−1.9
int. uov-V	5.4×10^{19} yr	4.0×10^{29}	$2^{201.3}$	+128.3	$2^{220.9}$	−2.4
wedge uov-Ip	7.0×10^3 yr	5.8×10^5	$2^{82.6}$	**+9.6**	$2^{141.0}$	+0.3
wedge uov-Is	9.9×10^{10} yr	3.1×10^{16}	$2^{135.3}$	+62.3	$2^{177.4}$	+1.5
wedge uov-III	7.1×10^{16} yr	1.9×10^{25}	$2^{179.8}$	+106.8	$2^{206.4}$	−0.6
wedge uov-V	7.9×10^{26} yr	2.2×10^{40}	$2^{254.9}$	+181.9	$2^{256.8}$	−1.0

Exactly one cell can buy its way to a year: the intersection attack on uov-Ip, at $k' \approx 49$ and $2^{62.2}$ bits — about 3,600 racks, roughly 0.6 EiB, a hyperscaler campus, and obtainable. No other cell can. Ran's wedge attack on the same parameter set already needs $2^{82.6}$ bits, nearly 800 times all the DRAM ever manufactured, and the higher sets stop being engineering at all: before any blocking, the cooled spheres for wedge uov-III and uov-V have *diameters* of 26 km and 8,400 km — the latter two-thirds the diameter of the Earth — at which point the sphere, the cooling, and the light-crossing are all euphemisms. Distance itself turns hostile at that scale: past ∼80 – 100 km every fiber span needs amplification (Section 9.2), so one crossing of the wedge uov-V machine costs 2^{15} – 2^{18} gate-energies per bit in amplifier energy alone, dwarfing the transceiver toll it was granted. We record this section as an independent constraint rather than as part of the gate accounting: it uses only c, the matrix dimension, transceiver physics, and commodity-hardware densities pushed a further 8× in the attacker's favor, so it stands whether or not one accepts the memory-cost model of Section 4 and Section 9.

A bleak physical lower bound on latency. We give our most bleak physics-based lower bound for the latency of performing a matrix-vector product. Suppose the vector v of length N is stored in a physical memory shaped like a cube of side length L, and consisting of $M = N \log_2(q)$ bits. Consider any protocol that computes Av in place. We assume the matrix A *mixes well*: however one slices the cube in half, the resulting split $v = (v_1, v_2)$ and block decomposition $\left(\begin{smallmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{smallmatrix}\right)$ of the acting matrix always has off-diagonal blocks A_{12} and A_{21} of rank $\Theta(N)$. In other words, no geometric cut separates the data into two halves that barely talk to each other. Two physical constraints now bound the latency T. First, the mixing property means that outputs on one side of the cube depend on inputs on the other, so information must travel distances of order L. Since information does not travel faster than the speed of light, we have $T \geq L/c$. Second, cut the cube down the middle. The high rank of A_{12} and A_{21} forces roughly M bits of information to cross the cut. The cut has area L^2, so if X is the maximum information flux per unit time and area, then $T \geq M/(XL^2)$. The two bounds pull in opposite directions:

a smaller cube helps information travel across the length of the cube faster, but chokes the bandwidth through the cut. Balancing them at $L^\star = (cM/X)^{1/3}$ gives the size-independent bound

$$T \;\geq\; \min_{L>0} \max\left\{ \frac{L}{c},\; \frac{M}{XL^2} \right\} \;=\; \left(\frac{M}{c^2 X} \right)^{1/3}.$$

For the intersection attack on the Ip parameter set, we have $M = 2^{53.6}$ bytes $= 2^{56.6} \approx 1.1 \times 10^{17}$ bits. An optimistic area bandwidth of optic fiber is $X = 4 \times 10^{19}\ \mathrm{bit\,s^{-1}m^{-2}}$, this gives $T \gtrsim 3.1 \times 10^{-7}\,\mathrm{s}$, attained around $L^\star \approx 94\ m$. This means that it is phisically not possible to do a single in-place multiplication of a well-mixing matrix at this scale below a third of a microsecond, unless one can improve on the communication throughput of optic fiber.

Since the attack requires $3N = 3 \times 2^{53.6}$ sequential matrix-vector products, this gives a lower bound of $3 \times 2^{53.6} \times 3.1 \times 10^{-7} s = 390$ years on the runtime of the attack, regardless of how much money and resources are available to the attacker.

9 On the cost of moving bits: does fiber make transmission *much* cheaper than 2^{-5}?

The memory-cost model prices a remote access at $2^{-5}\ (b + \log N)\ \sqrt{M}$: the constant $C_2 = 2^{-5}$ was chosen by the NTRU Prime team and reused by the Rainbow team and by the Crypto 2022 paper. In other words, everyone in this discussion has agreed that it is not too unreasonable. The informal claim we want to examine — never, as far as we can tell, committed to print — is that fiber-optic links between nodes make transmission *much* cheaper than $C_2 = 2^{-5}$ indicates, cheaply enough to erode the security margin the memory model buys back.

We do not dispute that fiber may push the constant *somewhat* below 2^{-5}. The question is whether it can push it *much* below — by the several bits that a successful attack argument might need — and our claim is that it cannot. Two things bound it:

1. the exponent on M is untouched by the choice of medium (Section 9.1), and 2. the constant itself is not conservative against the attacker (it was estimated based on the chip physics of one decade ago, which has not changed substantially), with concrete physics capping how far fiber can move it and one term that arguably moves it the wrong way (Section 9.2).

Throughout, the reference point is that the useful work — an $\mathbb{F}_{2^8}$ multiply — costs $\sim 2^6$ – 2^7 gates, so a transmission cost has to be judged against *that*, not against a long copper run.

9.1 The exponent is geometric, and no medium changes it

Whatever fiber wins is a constant factor, because the $\sqrt{M}$ itself is a statement about graphs embedded in physical space, not about signaling.

Distance. Bits occupy space at bounded density, so M bits fill a region of linear extent $\sim \sqrt{M}$ (2D) or $M^{1/3}$ (3D), and two random locations are $\Theta(\sqrt{M})$ apart. Fiber propagates at $\sim \frac{2}{3}c$ — fast, finite, still $\propto \sqrt{M}$. (Latency can be pipelined away, but the model charges resource·time, not wall-clock: you pay for the pipe for the whole duration whether or not you hide the latency behind useful work.)

Bisection. This is the medium-independent part — the Thompson VLSI / AT^2 bound. Any split of the memory into two halves has a waist of linear size $\sim \sqrt{M}$; the channels crossing it number (waist)/(pitch), so wT $\geq$ B with $w \leq \sqrt{M}/\lambda$. Random access to an M-word memory forces $\Theta(M)$ bits across the waist, hence $\Theta(\sqrt{M})$ time. Thompson states this as $A \geq \omega^2/4$ (area at least a quarter of the squared bisection width), proven by crossing-number / wire-area arguments in which the medium never appears. A general-audience entry point is the Wikipedia **bisection bandwidth** article, which records the same $\sqrt{n}$ cut scaling.

So the debate is legitimately about C_2, and only about C_2.

9.2 How much can fiber actually move the constant?

Not much, and not obviously downward for the part that matters.

Pitch does not help — it hurts. What fiber buys is low loss over distance, not more channels through a cross-section. A single-mode fiber is $\sim$125 μm of cladding, enormous next to a $\sim$10 nm on-chip wire. So fiber has a *worse* channel pitch than fine copper: it does not widen the $\Theta(\sqrt{M})$ waist, and once you are waist-limited, fatter individual pipes do not add throughput. (WDM buys back several channels per fiber, but that is a bounded multiplier on the constant, not a change in scaling.) The place fiber legitimately helps is the *length* of the genuinely long links — a coefficient improvement on the residual remote traffic, which is real but presumably bounded.

Energy arguably pushes the optical fraction's constant the wrong way. An optical link's cost is dominated by the transceivers — modulator/laser plus photodiode + TIA + SerDes — a fixed per-bit tax of order a picojoule for short reach that does *not* vanish as distance shrinks. A local gate op is femtojoule-scale. So one optical hop runs $\sim 10^2 - -10^3$ gate-op energies per bit: for the fraction of traffic that actually goes optical, the transceiver overhead is comparable to or larger than the $\mathbb{F}_{2^8}$ multiply it serves. Fiber is cheap relative to long copper; it is not cheap relative to local compute in a small field. If anything, moving traffic onto optics raises the effective constant on that traffic rather than lowering it. Purchasable optics today sits well above even that range: 800G pluggable modules run 10 – 15 pJ/bit and production co-packaged optics 5 – 8 pJ/bit — $2^{12.5}$ – $2^{14.5}$ gate-energies per bit — so the 10^2 – 10^3 used throughout corresponds to roadmap silicon photonics, not to parts on a shelf. And distance in fiber is nearly free only until amplification: passive glass at 0.2 dB/km adds $\sim 10^{-19}$ J per bit per meter, but past $\sim$80 – 100 km every span needs an amplifier at $\sim$0.5 – 2 pJ/bit, of order tens of attojoules per bit per meter — a genuinely per-distance term that Section 8 charges against the kilometer-scale machines.

2^{-5} is not defender-generous. The constant was chosen on the chip physics of 2016–17. To reduce the security margin materially, fiber would have to buy several *bits* — a factor of tens — off a constant where physics barely moved. Between the pitch penalty and the transceiver floor, there is no headroom for a swing of that size; what remains is a sub-bit, coefficient-level adjustment to the residual remote term.

9.3 A sanity check from real bulk transport

The cheapest per-bit transmission technology we actually have is not a faster link — it is shipping the disks, and even that is cheap only in a way the attack cannot use.

- **Google Transfer Appliance.** Their own comparison: at 100 Mbps, 300 TB takes about 9 months to upload; the appliance captures 300 TB in under 25 days, with the data typically in cloud storage ∼10 business days later, consuming no outbound bandwidth. Local ingest is 10/40/100 GbE.
- **AWS Snowball.** The Storage Optimized device holds 210 TB; on-demand job fees are about USD 1,800 (≤100 TB) or USD 3,200 (101–210 TB) including 15 days, then USD 250/day, plus shipping. Data in is free per GB; data out is billed (their example uses USD 0.05/GB).

The lesson for C_2: the cheapest bit-movement we know is a **latency–bandwidth trade** — excellent per bit and bits per second bought by accepting multi-week delay and moving the data *once*, sequentially, as a block. That is the opposite of the Wiedemann inner loop, which needs $\Theta(V)$ *random-access* touches repeated ∼3V times. You cannot ship a box of disks once and thereby service 10^{12} low-latency random reads. So even the most favorable real transmission technology makes cheapness contingent on sequential, one-shot, high-latency access — exactly the access pattern the attack does *not* have, and exactly what the $\sqrt{M}$ term is pricing.

9.4 What we concede, and the one-line version

- Fiber (and photonic interconnect generally) is a legitimate reason to argue C_2 *somewhat* below 2^{-5} on the residual long-distance traffic, and to prefer the 3D constant over the 2D one. That is a sub-bit to low-order-bit adjustment.
- For the structured, mostly-local traffic the hashing strategy produces, fiber is largely irrelevant; it touches only the remote remainder.
- None of this bears on the fan-out established for the wedge/intersection partitioning (the $\Theta(h^2)$ cluster count), which is a count of clusters that must fire and is orthogonal to per-hop cost.

In one line: fiber can move the *constant* C_2, not the $\sqrt{M}$ exponent — the exponent is geometry (distance $\Theta(\sqrt{M})$, waist $\Theta(\sqrt{M})$), and fiber's pitch is worse than copper's while its transceiver energy floor is higher than local compute. So 2^{-5} can be argued down by a bounded, sub-bit amount, not by the several bits the "much cheaper" claim needs. The cheapest real transport we have — shipping disks — agrees from the other side: cheap only when sequential, one-shot, and slow, which the random-access Wiedemann loop cannot exploit.

10 Summary of assembled security estimates

The table collects the whole accounting in one place, for both structural attacks side by side, with the direct forgery attack alongside and the perturbed sets of Section 5.2 below the rule. Each cost cell reads X_F (X_R): the Furue–Ikematsu intersection attack first, Ran's wedge attack in parentheses. The three cost columns are cumulative: **RAM** charges only field operations and prices memory at zero (the attacker's own accounting); **FLY** adds the cost of building the Macaulay matrix on the fly — address generation at the tightened $0.61 \cdot D(9L-3)$ count of

Section 2.2, amortized by $1/n'_r$ — but still leaves memory free; **MEM** adds the memory-access term on top, all costs counted, 2-D at $\rho_m = 1/n'_r$. All entries are $\log_2$ gate-equivalents at $h = 11$; † marks a figure below the NIST-required threshold. We show Furue first not because it is more likely correct — the opposite, if anything: Ran's wedge attack is the more widely accepted of the two, while the Furue intersection attack is the aggressive outlier whose validity is not established (Section 5) — but because it is the more demanding number, and we would rather answer the harder question. Ran's o' is not quoted from him: at $\ell = 1$ over $p = 2$ his attack is the reconciliation cell of the truncated-ring framework with $k = o - o'$ and $D = o'$, and our own search returns his 18/22/26/32 unprompted (Section 5.2).

scheme	design	NIST	RAM_F (RAM_R)	FLY_F (FLY_R)	MEM_F (MEM_R)	RAM_X	MEM_X
uov-Ip	145	143	128.4† (140.7†)	129.1† (142.6†)	141.3† (157.4)	145.6	151.1
uov-Is	143	143	158.5 (175.9)	160.4 (179.7)	181.1 (203.7)	162.5	171.0
uov-III	218	207	181.5† (206.9†)	182.4† (209.5)	207.5 (240.3)	217.9	241.5
uov-V	278	272	223.3† (257.8†)	224.2† (260.7†)	259.6† (304.0)	277.9	314.1
uov-Ip#	—	143	137.6† (148.0)	138.4† (150.1)	**152.9** (166.6)	145.6	151.1
uov-III#	—	207	189.3† (213.0)	190.2† (215.6)	**217.2** (247.9)	217.9	241.5
uov-V#	—	272	246.7† (271.2†)	248.0† (274.3)	**289.1** (320.8)	277.9	314.1

RAM_X and MEM_X are the direct (FXL) forgery attack, priced at zero memory and with memory charged: the round-2 specification's own §4.2 model, min'd against Hashimoto's algorithm, and then re-minimized over the guess exponent under the memory metric — because unlike the structural attacks FXL *has* that dial, and uses it. That is why its memory correction is only 5.5 to 36 bits where the structural attacks take 12 to 46, and why its vectors stay at $2^{38.5}$ – $2^{97.0}$ bits while theirs run to $2^{127.5}$. The bottom three rows are the perturbed sets of Section 5.2, at which every column clears.

Three things the table is meant to make unmissable.

(1) Building on the fly is a real cost, routinely underplayed — but it is not, by itself, decisive. Read the FLY column against RAM. Address generation adds 0.7 to 3.8 bits: $2^{-0.7}$ to $2^{3.7}$ times a field multiply after per-block reuse, and $2^{4.8}$ to $2^{8.0}$ before it. Against the "$< 2^7$ gates" and "sub-0.1%" of Appendix B.1 that is 9 to 14 bits of understatement, so the negligibility claim does not survive. Charging the same term at every cluster in the fan-out ball would put it at 2^{11} – 2^{14} times a field multiply, and that is wrong (Section 2.4): the hash is a function, so the monomials are partitioned, not replicated. FLY alone does not lift uov-Ip over threshold even against Ran's wedge attack — it reaches $2^{142.6}$ against a requirement of 143. The on-the-fly term is a backstop worth a couple of bits, not a rescue.

(2) Memory costs, and more than the clean $h/2$ the hashing strategy promises, because the partition is not uniform. The MEM column adds the access term, and it is what carries every verdict. For Ran it lifts all four sets far clear (+14.4 to +60.7), and still clears them in the physically unavailable 3-D model (+5.5 to +48.7). For Furue it lifts uov-Is and — by half a bit — uov-III. The access-weighted 3/2-moment computation of Section 4 confirms the partition penalty at 1.87 to 2.33 bits, so the realized saving is short of $h/2$ by that much: the cut is not perfect, and the imperfection runs against the attacker.

(3) The cells that stay below — uov-Ip and uov-V under Furue — are precisely where the burden of proof lies, and it lies with the attacker. The Furue intersection attack reaches those numbers only under a chain of choices the paper leaves implicit: that memory is free, that the matrix costs nothing to regenerate, that the semi-regular solving degree

D is exactly as conjectured (his own §4.4 shows it missing by a degree on a test instance, and one extra degree is +5.6 bits on the squared cost — more than uov-Ip's deficit), and that no mid-range *working* instance need be exhibited before the estimate is treated as a lower bound. It is not the defender's job to grant all of these and then find the remaining bits; it is the proposer's job to close them. A complexity that lowers a NIST security claim should arrive with its omitted costs accounted and its solving degree demonstrated on solvable instances — and until it does, the conservative reading is the one that stands. Leaving those details out and later retreating to "that is not what we meant" shifts a burden that was never the defender's to carry. The three rows below the rule are what we would do if that burden is nonetheless placed on us: uov-Ip#, uov-III# and uov-V# clear every column of this table, the binding attack by +8.1, +10.2 and +17.1 bits on MEM — the forgery at level I, the intersection at III and V and the wedge by +5.0 and +6.0 on RAM alone at levels I and III, at a cost of +7.2%, +2.2% and +5.0% on the expanded public key and nothing whatever on the compressed one.

Contents

1 Historical comment on a comment **2**

2 Building (and using) the matrix on the fly **2**

2.1 Shared-support rows, and an identity that fixes the reuse factor 2

2.2 A minimalist cost per address generation 3

2.3 Partitioning cost in the favorable case of Support-Minors 4

2.4 Why intersection / wedge attacks are worse: the fan-out is quadratic in h 4

2.5 Row weight and the Ran encoding 6

3 On doing things cheaply by random seed **6**

3.1 Squeezing harder — many "random" picks from one call 6

4 Assembling the wedge-attack cost with the memory term **8**

4.1 Putting it together, and why the partition is not uniform 8

4.2 Check-work: uov-Is, the set the wedge attack does not reach 10

5 The Furue–Ikematsu intersection attack, and the all-in cost **10**

5.1 Independent reproduction of the optimizer, and where its accepted cells sit . . . 12

5.2 Proposed replacement parameters: uov-Ip#, uov-III# and uov-V# 13

5.3 The alternative if the memory argument is rejected 16

6 Block Wiedemann **17**

7 Why h cannot simply be raised: the units stop being closely coupled **18**

8 Latency and distance: a time–memory trade-off like AT^2 **19**

9 On the cost of moving bits: does fiber make transmission *much* cheaper than 2^{-5}? **22**

9.1 The exponent is geometric, and no medium changes it 22

9.2 How much can fiber actually move the constant? 23

9.3 A sanity check from real bulk transport 23

9.4 What we concede, and the one-line version 24

10 Summary of assembled security estimates **24**

Parameters used **29**

Reproducibility **29**

References and sources **30**

Acknowledgments **31**

Parameters used

UOV round-2 recommended sets: uov-Ip (q=256, n=112, o=m=44, v=68); uov-Is (q=16, n=160, m=64, v=96); uov-III (q=256, n=184, m=72, v=112); uov-V (q=256, n=244, m=96, v=148). Wedge operating parameter and matrix: square dimension $N = \binom{v+o'}{v}$, row weight $\binom{v+2}{2}$, with $o' = 18/22/26/32$ for Ip/Is/III/V and 19/27/34 for Ip#/III#/V#; these are re-derived as the $\ell = 1$ reconciliation optima with $o - k = D$, not quoted from Ran [ePrint 2025/1143]. Intersection: $N_{col} = \binom{n-k}{D} - \binom{2o-v-k}{D} + 1$ with $(\ell, k, D) = (1,7,13)/(1,16,16)/(1,14,18)/(1,22,22)$; these (ℓ, k, D) are re-derived, not quoted (Section 5.1). Proposed replacements of Section 5.2: uov-Ip# (256,116,44), uov-III# (256,186,72), uov-V# (256,250,96), with intersection (ℓ, k, D) = (1,2,14)/(1,11,19)/(1,13,25) and reconciliation (1,25,19)/(1,45,27)/(1,62,34), the latter being the wedge cells above; the superseded candidate of Section 5.3 is (256,128,48) at (1,0,16). Direct attack: UOV round-2 specification §4.2 (Thomae–Wolf to m-1 equations in m-1 variables, then hybrid Wiedemann XL) and Hashimoto's algorithm in the form of the SNOVA round-2 specification §5.1. Field multiply $2(\log_2 q)^2 + \log_2 q$ and L-bit add $9L - 3$ gates as in the cost model of the Crypto 2022 paper and its [cites 16,40]. Hash parameter $h = 11$ with mean overlap $a = 2$; $C_2 = 2^{-5}$.

Reproducibility

Every number in this note is regenerated from the parameters by `pwxl_UOV_costs.py`, which uses only the Python standard library and runs unmodified under `sage -python`. Hilbert-series arithmetic is exact integer convolution and exact rational division. The script models both attacks and asserts 58 invariants, among them: reproduction of Furue's own RAM-model gate counts to within 1 bit; the identity $(R/T) \cdot n'_r = m$ on every instance of both attacks; reproduction of the MEM figures of Section 4 and Section 5; the closed form $\delta(p)$ against the measured penalty slope, within 0.02 bit; Vandermonde for the bucket sizes; the partition penalty equal to 0 at $h = 0$ and non-negative for all h, by Jensen; the blocking optimum $k'^* = A/c$ against the numeric minimum of $f(k')$; exhaustive verification over all D-subsets that the colex bound S_j is exact; and the Section 8 machine — 76 racks, the 3.0 m sphere, the 13-year latency floor, and TRANS within -2.4 to $+1.5$ bits of RAM on all eight cells. The parameter search of Section 5.1, Section 5.2 and Section 5.3 lives in the same script and takes (q, p, e, n, m) rather than a table of stored optima, asserting 31 further invariants: every filled cell and every blank of Table 1 of [ePrint 2026/298], all eight cells of its Table 3 with the published (ℓ, k, D), the figures of Section 5.3, that $k = 0$ is the unique $\ell = 1$ cell there, that each accepted cell satisfies $D = \alpha(p - 1)$ with $[G]_D < 0 < [G]_{D-1}$, and that Ran's published o' are recovered as the ℓ = 1 reconciliation optima on all four round-2 sets. The direct-attack figures of Section 5.2 assert 21 more: the Direct column of Table 5 of the UOV round-2 specification to within a bit, all nine $l \leq 4$ cells of the Direct column of Table 8 of the SNOVA round-2 specification exactly, and the n-thresholds at which each Hashimoto cell becomes admissible. the memory-charged RAM_X/MEM_X columns are computed re-minimizing over the guess exponent under each metric,

References and sources

Attacks and parameter sets.

- L. Ran, *Wedges, oil, and vinegar — an analysis of UOV in the exterior algebra.* EUROCRYPT 2026; Cryptology ePrint Archive 2025/1143. https://eprint.iacr.org/2025/1143
- H. Furue and Y. Ikematsu, *Key recovery attacks on UOV using p^ℓ – truncated polynomial rings.* Cryptology ePrint Archive 2026/298. https://eprint.iacr.org/2026/298
- J. Baena, P. Briaud, D. Cabarcas, R. Perlner, D. Smith-Tone, J. Verbel, *Improving Support-Minors rank attacks: applications to GeMSS and Rainbow.* CRYPTO 2022; Cryptology ePrint Archive 2021/1677. https://eprint.iacr.org/2021/1677
- UOV — specification document, round 2 (NIST additional signatures), 2025. https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures
- W. Beullens, *Improved cryptanalysis of UOV and Rainbow* (intersection attack). EUROCRYPT 2021. https://eprint.iacr.org/2020/1343

XL / Wiedemann and the on-the-fly cost.

- C.-M. Cheng, T. Chou, R. Niederhagen, B.-Y. Yang, *Solving quadratic equations with XL on parallel architectures.* CHES 2012, LNCS 7428, 356–373; extended version ePrint 2016/412. https://eprint.iacr.org/2016/412
- R. Niederhagen, *Parallel Cryptanalysis.* Ph.D. thesis, TU Eindhoven, 2012 (§4.5.3 on tall-and-thin Macaulay matrices and row survival). https://polycephaly.org/thesis/index.shtml
- T. Chou, R. Niederhagen et al., *Fast exhaustive search for quadratic systems in F_2 on FPGAs* (extended version); the RIVYERA Spartan-6 machine. https://tungchou.github.io/papers/forcemp-fpga.pdf
- D. H. Wiedemann, *Solving sparse linear equations over finite fields.* IEEE Trans. Inf. Theory 32(1):54–62, 1986.
- D. Coppersmith, *Solving homogeneous linear equations over GF(2) via the block Wiedemann algorithm.* Math. Comp. 62(205):333–350, 1994.
- M. Bardet, J.-C. Faugère, B. Salvy, *Complexity of Gröbner basis computation for semi-regular overdetermined sequences over F_2.* INRIA RR-5049, 2003. https://inria.hal.science/inria-00071534

Cost model, gate counts, thresholds.

- D. J. Bernstein, C. Chuengsatiansup, T. Lange, C. van Vredendaal, *NTRU Prime: reducing attack surface at low cost* — origin of the memory-access cost model and the constant $C_2 = 2^{-5}$ reused by the Rainbow and Crypto 2022 analyses. SAC 2017. https://ntruprime.cr.yp.to/ntruprime-20170816.pdf

- NIST, *Security (evaluation criteria)* — classical gate-count thresholds $2^{143}/2^{207}/2^{272}$ for categories I/III/V (AES-128/192/256 key search). `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria)`
- J. Boyar, R. Peralta, *A new combinational logic minimization technique with applications to cryptology* (AES S-box in 32 AND + 83 XOR). SEA 2010; ePrint 2009/191. `https://eprint.iacr.org/2009/191`

Physical limits and transport.

- C. D. Thompson, *Area–time complexity for VLSI*. STOC 1979 ($AT^2 \gtrsim N^2/16$; layout-area/bisection lemma $A \geq \omega^2/4$). `https://arxiv.org/pdf/1602.04034`
- *Bisection bandwidth* (general-audience; $\sqrt{n}$ cut scaling): `https://en.wikipedia.org/wiki/Bisection_bandwidth`
- SciEngines RIVYERA S6-LX150 — up to 256 Spartan-6 LX150 FPGAs per single system; beyond one chassis, a "cluster ... interconnected by network." `https://www.sciengines.com/wp-content/uploads/RIVYERA_S6_LX150_REV38_DATASHEET.pdf`
- AWS Snowball (Storage Optimized, 210 TB; job pricing): `https://aws.amazon.com/snowball/pricing/`
- Google Transfer Appliance (bulk one-shot ingest speeds): `https://docs.cloud.google.com/transfer-appliance/docs/4.0/overview#transfer-speeds`

Bracketed reference tags in the text ([16,40] for the memory model, [cite 16] for Cheng–Chou–Niederhagen–Yang, [cite 40] for the Ding et al. "matrix-on-the-fly underestimates the complexity" remark) point into the Crypto 2022 reference list; the corresponding works are those listed above.

Acknowledgments

The writing of this note was grammar- and spell-checked with Gemini 3.1 Pro, Claude Opus 4.8, and Grammarly; the costing (and its modeling) was additionally checked by Claude Opus 4.8 in High Thinking mode, alongside the authors.