Z6 Admission Layer — Benchmark and Reproducibility Report

Stateless denial-of-service protection for ML-KEM-768 endpoints

Measurement date 2026-07-19 · Node v22.22.2 · single core · ML-KEM-768 via @noble/post-quantum

Purpose

This document exists so that every figure published at secz6.ooooooooo.se can be independently regenerated. It states the methodology, the exact commands, and the raw output. Where a measurement disagrees with a previously published figure, both are given. Nothing here is a security proof: search hardness is measured against five attack strategies, and confidentiality is provided by ML-KEM-768 (NIST FIPS 203) used unmodified, not by this layer.

1. What is being measured

ML-KEM-768 decapsulation is expensive and any client can trigger it by sending a 1,088-byte ciphertext. The admission layer places a 24-byte proof-of-work token in front of that operation. Targets derive from SHA-256 of the token itself, so no element can be repaired locally and a valid token must be found by search. Difficulty is 2^a · 3^b, set by a constraints checked mod 2 and b mod 3 — a consequence of Z/6Z factoring as Z/2Z × Z/3Z.

2. ML-KEM-768 baseline

Operation	Time	Notes
Key generation	751.98 us	
Encapsulation	729.32 us	client side
Decapsulation	**791.35 us**	**the cost admission gates**
Public key	1,184 B	
Ciphertext	1,088 B	triggers decapsulation
Admission token	24 B	2.2% overhead

Discrepancy disclosed: the gateway's own earlier benchmark measured decapsulation at 585 us on different silicon. Every ratio in this document uses 791.35 us, the slower figure, which is the conservative direction — it makes the advantage claims harder to reach, not easier.

3. Verification cost is flat in difficulty

a	b	Difficulty	Verify	vs decapsulation
1	1	6	1.636 us	484x cheaper
2	1	12	1.516 us	522x cheaper
3	2	72	1.610 us	491x cheaper
4	3	432	1.591 us	497x cheaper
6	4	5,184	1.608 us	492x cheaper
8	5	62,208	1.544 us	513x cheaper

1.516–1.636 us across a 10,368× difficulty range. This is the property the design depends on: raising the price on an attacker costs the defender nothing.

4. Grind cost matches prediction

a	b	Difficulty	Predicted 0.693d	Measured	Ratio	Client p50	p90

2	1	12	8	10	1.20	0.03 ms	0.12 ms
3	2	72	50	50	1.00	0.15 ms	0.59 ms
4	2	144	100	90	0.90	0.29 ms	1.39 ms
3	3	216	150	168	1.12	0.51 ms	2.16 ms
4	3	432	299	267	0.89	1.02 ms	3.82 ms
5	3	864	599	578	0.97	2.09 ms	6.61 ms

Ratios 0.89–1.20 across a 72× range. Search cost tracks difficulty with no structural shortcut emerging as the space grows.

5. Random-token admission rate

Setting	Designed	Measured	Samples
a=2, b=1	1 in 12	1 in 12	200,000
a=3, b=2	1 in 72	1 in 72	200,000
a=3, b=3	1 in 216	1 in 223	200,000

6. Attack surface

Five strategies were run against the construction. Each targets a specific way the structure could leak. None beat random search.

Method	k=3	k=4	Wins if
Brute force	1.06x	0.87x	(baseline)
Simulated annealing	**0.77x**	**1.74x**	**landscape has a gradient**
Coordinate solving	1.50x	1.14x	groups are independent
Meet-in-the-middle	0.99x	—	targets are key-independent
Biased sampling	1.15x	—	the hash is biased

Annealing is the decisive control: it wins whenever a constraint landscape has structure to climb. It moves from 0.77× at k=3 to 1.74× at k=4, so as the space grows six-fold it falls further behind brute force. That is the signature of a gradient-free landscape — every edit rerolls the SHA-256 targets, leaving nothing to descend.

7. Difficulty ladder granularity

Ladder	Rungs below 10^7	Worst gap	Mean overshoot
Powers of 2 (hashcash)	24	2.00x	1.190x
Powers of 6	9	6.00x	1.606x
2^a * 3^b (Z/2Z x Z/3Z)	**190**	**2.00x**	**1.031x**

Under load-adaptive difficulty, overshoot is grind that honest clients pay unnecessarily. The mixed-radix ladder tracks a load-driven target to 3% where binary puzzles overshoot by 19%.

8. Quantum sizing

Admission is a search problem, so Grover applies and each rung is worth half its classical value: a=4,b=4 gives a quantum attacker what a=2,b=2 gives a classical one. The oracle was transpiled for ibm_marrakesh (native cz, rz, sx, x) under forced layout.

Component	Two-qubit gates	Share of oracle
Z/6Z constraint check, per group	~580	0.02%
SHA-256 quantum oracle	**12,582,912**	**99.98%**
Grover at a=4,b=4	1.28 x 10^10 total	1,017 oracle calls

Because the oracle is almost entirely SHA-256, quantum attack cost rests on a primitive with published resource estimates rather than on anything bespoke to this design.

Methodology warning: at optimization_level=3 Qiskit absorbs permutations into qubit allocation and reports oracle depth 416 against 717 under forced layout. Any resource estimate that does not pin the layout understates cost by roughly 1.7×.

9. Deployment sizing

Cores that must be provisioned in advance to absorb a sustained flood, since headroom cannot be acquired once an attack begins. Attack rates are published figures.

Scenario	Flood rate	Unprotected	With layer	Standby cost delta
Small SaaS	5,000 rps	4 cores	1 core	$88 / month
Retail bank	500,000 rps	396 cores	1 core	$11,534 / month
Hyper-volumetric	**100,000,000 rps**	**79,135 cores**	**150 cores**	**$2,306,362 / month**

Largest on record	398,000,000 rps	314,958 cores	597 cores	$9,179,342 / month

At $0.04 per vCPU-hour, 730 hours per month. In August 2023 Cloudflare mitigated 89 separate HTTP attacks exceeding 100 million requests per second; in October 2023 Google mitigated one peaking at 398 million. At those rates the unprotected requirement is not a budget decision — the headroom does not exist at any price, which is why the endpoint fails.

10. Reproducing every figure

All scripts are dependency-light and run on stock Node and Python.

```
# sections 2-5, 7 : benchmark suite
node bench.mjs

# section 6 : attack harness, five strategies, live endpoint check
node attack.mjs --k 4 --reps 200
node attack.mjs --k 5 --reps 500        # heavier run

# section 8 : Grover oracle transpilation (no quantum hardware time)
python grover_oracle.py --backend ibm_marrakesh

# section 9 : deployment sizing
#   open secz6.ooooooooooo.se and use the calculator, or compute directly:
#   cores = ceil(rate * 791.35e-6)   unprotected
#   cores = ceil(rate *   1.5e-6)    with layer

# live verification against the deployed gateway
curl 'https://sec6z.ooooooooooo.se/'
```

11. Standing challenge

Produce an admissible token for the current epoch in materially fewer than 0.693 · 2^a · 3^b attempts. A method that beats this — by exploiting the constraint structure, the group sums, or the packing — breaks the admission layer. The harness flags any result below 0.5× baseline as a break.

12. Scope and limitations

• Confidentiality is provided by ML-KEM-768 used unmodified. This layer performs admission control and HKDF context binding only. It is not a cipher and does not replace one.

• Search hardness is established by measurement against five attack strategies, not by proof. The construction has not undergone external cryptanalysis; the standing challenge exists to invite it.

• Per-request nonce binding is mandatory. Under per-epoch binding an attacker grinds once and replays; the cost advantage crosses unity at 226 replays and inverts beyond it.

• Timings are single-core on one machine and will vary with hardware. Decapsulation in particular was measured at both 585 us and 791.35 us on different silicon.