Pretzel: Compact Lattice Signatures from Module-LWE and Module-SIS

Alberto Centelles Hidalgo
Imperial College London
a16z frontier research

Justin Thaler
Georgetown University
a16z frontier research

Markos Georghiades
a16z frontier research

Abstract

We present Pretzel, a lattice-based signature scheme in the style of ML-DSA. Across parameter sets targeting NIST security categories 2, 3, and 5, its signatures are 52–59% smaller than ML-DSA's, it signs and verifies faster in all three categories, and its quantum security proof needs fewer assumptions. Across all three parameter sets, Pretzel produces smaller signatures than SHUTTLE (Ming et al., ASIACRYPT 2026), the recent state of the art for iterative rejection sampling from Module-LWE. At category 3, for example, Pretzel's 1361-byte signatures are 14% smaller than SHUTTLE's 1589 bytes. Unlike the quantum analyses of ML-DSA and SHUTTLE, Pretzel's security proof in the quantum random oracle model needs no self-target assumption: it relies only on Module-LWE and Module-SIS.

Like SHUTTLE, Pretzel builds on Gärtner's iterative rejection sampling (CRYPTO 2025). Its sampler keeps the response exactly on its target distribution after every update, which allows a substantially narrower mask than SHUTTLE's. Its verifier also enforces an additional bound on the response coefficients. This bound lets the security proof adapt Poremba's LWE-phase technique (ITCS 2023) to Module-LWE and rely on Module-LWE and Module-SIS alone.

1 Introduction

Signature size remains an obstacle to the transition to post-quantum cryptography. Among NIST's post-quantum signatures selected for standardization [20, 35, 36], the lattice-based ML-DSA (the standardized form of Dilithium [17]) and Falcon have the smallest signature sizes. ML-DSA signatures are 2420–4627 bytes, compared with 7856 to 49,856 bytes for SLH-DSA, which is based on hashes. In contrast, elliptic-curve signatures such as ECDSA and EdDSA are about 64 bytes, but they are not secure against quantum attacks [34].

Falcon and ML-DSA both build on well-studied lattice problems. Falcon's signatures are smaller than ML-DSA's. Historically, Falcon's signing has been harder to implement than ML-DSA's and has needed more memory, because it samples from a discrete Gaussian distribution with high-precision arithmetic. Recent work mitigates both drawbacks: NIST's planned FIPS 206 specifies the sampler in fixed point [37], and a recent implementation cuts Falcon's signing RAM by about two thirds [39]. Nonetheless, Falcon's security relies on the hardness of the small integer solution (SIS) problem over NTRU lattices, whereas ML-DSA's classical security proof rests solely on module learning with errors (Module-LWE) and module small integer solution (Module-SIS) [26,35]. For the standardized parameters, however, ML-DSA's quantum analysis still uses an additional self-target Module-SIS assumption [25,26].[1] Quantum proofs that avoid this assumption exist, but only for variants whose signatures are more than twice as large [25, 26].

Gärtner's iterative rejection sampling offers a route to smaller signatures within the Fiat–Shamir with rejection sampling framework introduced by Lyubashevsky [30, 31] and used by ML-DSA [22, 35]. Instead of hiding one large secret-dependent change behind a large random mask, the signer makes a sequence of small changes and hides each one.

Gärtner's concrete parameters use NTWE, a lattice problem combining features of NTRU and LWE [21]. NTWE introduces an additional short secret polynomial f, changing the public-key relation from $b = As + e$ to $fb = As + e$ modulo q. He also describes a Module-LWE variant with larger signatures and leaves an optimized implementation open [22, Section 5.2]. Lithium develops the NTWE construction into practical signatures, with 1618-byte signatures at the ML-DSA-65 security level, and inherits Gärtner's self-target assumption [46].

SHUTTLE [33] further develops the iterative sampling approach into a practical construction with Module-LWE public keys and integer-only sampling. It reports smaller signatures and faster verification than ML-DSA, but slower signing [33, Tables 2–3]. Its quantum analysis still requires an additional bimodal self-target Module-SIS assumption [33, Section 5].

SHUTTLE thus improves on ML-DSA in signature size and verification time, but not in signing time or in the assumptions of its quantum analysis. Roughly speaking, we seek a Pareto improvement over ML-DSA: smaller signatures, faster signing and faster verification, with a quantum security proof that needs fewer assumptions. Thus we ask:

Can we achieve smaller signatures than SHUTTLE, with both signing and verification faster than ML-DSA, while also avoiding their self-target assumptions?

1.1 Our results

We answer affirmatively in this work. For NIST category 3, Pretzel signatures are 1361 bytes, compared with 1589 bytes for SHUTTLE and 3309 bytes for ML-DSA-65. On Apple silicon (macOS) with 32-byte messages, Pretzel also signs with a prepared secret key about 7% faster on average and verifies about 44% faster than ML-DSA-65 in mean thread CPU time, including public-key preparation for verification. These timings include every retry. The signer is designed to run in constant time.

[1] Self-target Module-SIS asks for a short vector y and a message μ such that hashing μ together with Ay, for a public matrix A, gives part of y itself. Because the hash function is part of the problem, it is not a standard lattice problem.

Table 1: Pretzel versus ML-DSA at the three security targets. Time is shown as a percentage of ML-DSA's mean thread CPU time.

		Signature bytes		Time (% of ML-DSA)		
Target	Baseline	Pretzel	ML-DSA	Sign	Verify	Key gen.
Level 2	ML-DSA-44	1160	2420	91%	72%	147%
Level 3	ML-DSA-65	1361	3309	93%	56%	134%
Level 5	ML-DSA-87	1908	4627	95%	48%	99%

Measured on macOS (Apple M2 Pro), with 32-byte messages. Signing uses a prepared secret key. Times are means. Verification includes public-key preparation. The category-2, category-3 and category-5 rows use Tables 8, 7 and 11, respectively.

Pretzel also has parameter sets for NIST categories 2 and 5. We write Pretzel-(ℓ, m) for the parameter set whose secret key has ℓ secret and m error polynomials: Pretzel-(3,3), Pretzel-(4,4), and Pretzel-(5,5) target categories 2, 3, and 5, respectively. Table 1 compares all three with ML-DSA.

Our security proof in the quantum random oracle model (QROM), where adversaries may query the hash function in superposition, reduces forgery to Module-LWE and Module-SIS, with no self-target assumption (Section 2.4). The proof uses one additional verifier check: a weighted sum of the response coefficients must be small. As for ML-DSA, the reduction is not tight, so we set parameters from the best known attacks on these problems (Section 6).

The improvement over ML-DSA does not extend to key generation, which takes up to 1.47 times as long as ML-DSA's (Table 1). Pretzel's public keys are smaller than ML-DSA's at categories 3 and 5, but 14 bytes larger at category 2 (1326 bytes, compared with 1312 for ML-DSA-44).

1.2 Related work

Table 2 compares Pretzel with other Fiat–Shamir lattice signatures. Among them, G+G replaces rejection sampling by Gaussian convolution, so its protocol has no aborts [12], and Raccoon is designed so that signing can be masked against side-channel attacks at low cost, with 14,544-byte signatures at category 3 [9,10]. G+G also has a QROM proof from LWE alone, but it requires a much larger modulus than its proposed parameter sets use [12, Section 4.1]. Lithium's sets in the table target ordinary unforgeability; its sets for strong unforgeability have signatures of 1528, 1909 and 2817 bytes [46, Table 7]. Outside this family, the hash-and-sign scheme Falcon-1024 has 1280-byte signatures and 1793-byte public keys at category 5 [20, Table 3.3]; as noted above, its security relies on SIS over NTRU lattices rather than on Module-LWE and Module-SIS.

Comparison with SHUTTLE. Table 3 compares the published signature and public-key sizes across the parameter sets. Pretzel has smaller signatures in each pairing, while SHUTTLE has smaller public keys. Relative to ML-DSA, SHUTTLE reports signing 5–23% slower and verification 11–26% faster, while Pretzel signs 5–9% faster and verifies 28–52% faster. The two studies use different machines and different ML-DSA implementations, so these percentages do not compare the speeds of Pretzel and SHUTTLE directly. The smallest sets have different security targets: Pretzel-(3,3) targets category 2 and SHUTTLE-120 targets category 1, although both use ML-DSA-44 as a baseline.

At category 3, Pretzel-(4,4) uses dimensions $(\ell, m) = (4, 4)$: its polynomial matrix A_0 has four columns and four rows. The public vector $b = A_0 s_0 + e \bmod q$ has one polynomial per row, so the public key stores four polynomials, compared with SHUTTLE's three. Pretzel's quantum proof also relies on an additional Module-LWE problem whose secret has the same number of polynomials as the public key. Increasing this number from three to four raises the estimated cost of attacking that problem, but requires a larger public key (Section 6).

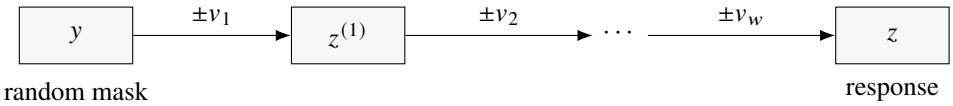
Table 2: Fiat–Shamir lattice signatures: hardness assumptions of the unforgeability proof and published sizes in bytes. ST-MSIS is self-target Module-SIS and BST-MSIS its bimodal variant; G+G's proof for its module parameters is in the classical random oracle model (ROM). Gärtner, Lithium and G+G state bit-security targets and are listed at the categories of the ML-DSA sets with matching targets; Gärtner's and G+G's sizes are estimates.

Category	Scheme	Hardness assumptions	Signature	Public key
3	Gärtner [22, Table 3]	NTWE, BST-MSIS	1184	1056
3	Pretzel-(4,4)	MLWE, MSIS	1361	1742
3	SHUTTLE-180 [33]	MLWE, BST-MSIS	1589	1376
3	Lithium-180 [46, Table 4]	NTWE, BST-MSIS	1618	1472
3	G+G-180 [12, Table 1]	MLWE, MSIS (ROM)	2143	1952
3	HAETAE-180 [5, Table 6]	MLWE, BST-MSIS	2349	1472
3	ML-DSA-65 [35]	MLWE, MSIS, ST-MSIS	3309	1952
5	Gärtner [22, Table 3]	NTWE, BST-MSIS	1694	1568
5	Pretzel-(5,5)	MLWE, MSIS	1908	2158
5	SHUTTLE-255 [33]	MLWE, BST-MSIS	2013	1696
5	Lithium-260 [46, Table 4]	NTWE, BST-MSIS	2348	2080
5	G+G-260 [12, Table 1]	MLWE, MSIS (ROM)	2804	2336
5	HAETAE-260 [5]	MLWE, BST-MSIS	2948	2080
5	ML-DSA-87 [35]	MLWE, MSIS, ST-MSIS	4627	2592

Table 3: Pretzel and SHUTTLE: sizes in bytes and time as a percentage of the ML-DSA implementation that each study measures against: the reference implementation on Intel for SHUTTLE [33, Tables 2–3], and AWS-LC's ML-DSA on macOS for Pretzel, with public-key preparation included in verification (Section 7). SHUTTLE does not report key-generation time. The timing percentages do not give a direct speed comparison between Pretzel and SHUTTLE.

Target	Scheme	Signature	Public key	Sign (%)	Verify (%)	Key gen. (%)
2	Pretzel-(3,3)	1160	1326	91	72	147
1	SHUTTLE-120	1175	992	106	87	–
3	Pretzel-(4,4)	1361	1742	93	56	134
3	SHUTTLE-180	1589	1376	105	89	–
5	Pretzel-(5,5)	1908	2158	95	48	99
5	SHUTTLE-255	2013	1696	123	74	–

Quantum security of Fiat–Shamir signatures. Kiltz, Lyubashevsky and Schaffner prove ML-DSA secure in the QROM under Module-LWE and SelfTargetMSIS, and without the latter only for enlarged parameters [26]. Liu and Zhandry, and Don, Fehr, Majenz and Schaffner, prove the Fiat–Shamir transformation secure in the QROM for protocols with suitable extraction properties [14, 28]. Jackson, Miller and Wang derive the quantum hardness of SelfTargetMSIS from Module-LWE for parameters larger than ML-DSA's [25]. The step from chosen-message to no-message security was tightened by Grilo et al. [23], corrected for signatures with aborts by Devevey et al. and Barbosa et al. [3, 11], and tightened further in [19]. These results concern that step only: the QROM bounds of ML-DSA, SHUTTLE and HAETAE keep their self-target assumptions.



Pretzel signing outline

1. Sample a fresh random mask y and salt ρ, and set $z \leftarrow y$.
2. Compute the commitment $U = Q(Hy + \delta)$ and the challenge $c = \mathrm{Fold}(\mathcal{H}(\mathsf{pk}, \mu, \rho, U))$. Select the positions j for which $c_j = 1$.
3. For each selected position j_i, let $v_i = X^{j_i} s$ and choose $z \leftarrow z - v_i$, $z \leftarrow z + v_i$, or restart from step 1. The probabilities are chosen to hide each secret-dependent update.
4. Check the final response and its encoding. If a check fails, restart; otherwise return the encoded signature.

Q replaces each coefficient by its interval number, as defined in Section 4.2. Here δ is the public offset, μ is the message digest, and pk is the public key. The function $\mathcal{H}$ hashes these inputs, and Fold converts its output into a challenge. Section 4.4 gives the details.

Figure 1: Iterative sampling: each v_i is a rotation of the short secret vector s. The arrows show a successful attempt. Like Gärtner's sampler, Pretzel may restart after any update in step 3. SHUTTLE never restarts during the updates: it always adds or subtracts, and restarts only if the final checks in step 4 fail [22, 33].

2 Technical Overview

2.1 Iterative sampling

In a Fiat–Shamir signature with rejection sampling, the response z is a random mask y plus a secret-dependent vector, and the mask must hide that vector. ML-DSA hides the whole vector at once, which requires a wide mask. Gärtner's iterative rejection sampling [22] instead adds or subtracts a sequence of small secret-dependent vectors, one for each nonzero challenge bit.

These updates leave the commitment unchanged. The commitment is a public value computed from Hy; the signer hashes it, together with the public key, a digest of the message and a fresh salt, to obtain the challenge. Here H is the public matrix, s is the secret vector, and q is the modulus. The matrix and vector have polynomial entries in X, with the ring and key relation defined in Section 4.1. They satisfy $Hs = 0 \pmod q$. Multiplication by X^j rotates coefficients with sign changes at the boundary. Every such rotation of s satisfies the same relation, so adding or subtracting it leaves Hz unchanged:

$$H(z \pm X^j s) = Hz \pm X^j Hs = Hz \pmod q.$$

Starting from $z = y$, the signer therefore preserves $Hz = Hy$ and hence the original commitment throughout the updates.

At each step, the signer chooses between adding, subtracting, and restarting with carefully chosen probabilities, so that, for a challenge chosen independently of the mask, after each successful step the response follows a fixed distribution that does not depend on the secret (Figure 1). For signing, where the challenge depends on the commitment, Appendix E compares the implementation with this ideal process, including failed attempts. This is Gärtner's key insight: hiding small updates one at a time permits a much narrower mask than hiding their combined effect at once. The narrower mask gives both SHUTTLE and Pretzel their size advantage over ML-DSA.

Pretzel adds two ideas to this procedure:

- **Restarts allow a narrower mask.** Keeping Gärtner's per-step restarts lets Pretzel use a narrower mask than SHUTTLE's. Together with Pretzel's response and commitment encoding, this gives smaller signatures in each pairing in Table 3.

- **A response bound for the security proof.** Recording a quantum forger's response can disturb its computation. Besides the usual norm checks, Pretzel's verifier bounds a weighted sum of $\log_2 \sec(2\pi x/q)$ over the coefficients x of the response the verifier reconstructs (the secant check). This sum controls the loss in the security bound when the proof accounts for that disturbance (Section 5.2). This helps obtain a QROM proof based only on Module-LWE and Module-SIS, with no self-target assumption.

2.2 Restarts allow a narrower mask

SHUTTLE removes restarts from the updates, allowing a small deviation from its intended response distribution. Its analysis controls this deviation by choosing a sufficiently wide mask [33, Section 3.2]. Pretzel instead retains Gärtner's restarts. Keeping restarts allows a narrower mask. For the 1,024 transmitted coefficients excluding the first response polynomial, z_0, Pretzel-(4,4) uses a narrower sampling distribution: its width, measured by the standard deviation of the corresponding continuous Gaussian, is about 46, compared with 82 for SHUTTLE-180. This makes large coefficients less common and helps reduce signature size.

Both signatures use 864 bytes to store a fixed number of low bits from each response coefficient. The remaining information is compressed separately. The remaining response and commitment information occupies 465 bytes in Pretzel and 677 bytes in SHUTTLE, counting the space left empty because signatures have a fixed length. This accounts for 212 of the 228 bytes saved. The other 16 bytes come from replacing SHUTTLE's 48-byte challenge seed with Pretzel's 32-byte salt [41].

Keeping restarts has two costs. First, as in Gärtner's scheme, whether an update restarts depends on the secret, so an implementation must not reveal when restarts happen. Second, restarts cost signing time: Pretzel-(4,4) needs about 1.21 attempts per signature on average. SHUTTLE never restarts during its updates, and its authors estimate that the final checks cause a restart with probability about 2^{-30} [33, Appendix A.5].

Pretzel keeps these costs low in three ways.

- **Precomputed inner products.** For a proposed rotation $v = X^j s$, the inner product $a = \langle z, v \rangle_W$ determines how the response's weighted squared norm E, defined in Section 4.1, changes:

$$E(z \pm v) = E(z) + E(s) \pm 2a.$$

  For rotations $v_i = X^i s$ and $v_j = X^j s$,

$$\langle z \pm v_i, v_j \rangle_W = \langle z, v_j \rangle_W \pm \langle v_i, v_j \rangle_W.$$

  Pretzel computes the initial values $\langle y, v_j \rangle_W$ for the fresh mask y at the start of each signing attempt. The second term comes from values precomputed once per key. Each earlier update thus contributes one integer addition or subtraction. This avoids computing it from the full response at every step, thus improving from Gärtner, Lithium and SHUTTLE [22, 33, 46].

- **Keeping restarts rare.** To preserve the required response distribution, the sampler uses two nonnegative weights, $\widehat{f}$ for subtraction and $\widehat{g}$ for addition (Section 4.3.2). Their sum can exceed one, so they cannot yet be probabilities. We divide both by a single public constant M that bounds their sum for every permitted key and response:

$$f = \frac{\widehat{f}}{M}, \qquad g = \frac{\widehat{g}}{M}.$$

  The remaining probability causes a restart:

$$p_{\text{restart}} = 1 - f - g.$$

To keep restarts rare, we want M as small as possible while still bounding $\widehat{f}+\widehat{g}$ for every permitted key and response. Pretzel finds this maximum and chooses a fraction just above it, allowing integer arithmetic with little additional restart probability. This maximum grows with the key's weighted squared norm $E(s)$, so, as in SHUTTLE, key generation accepts only keys with $E(s) \in [105{,}500, 230{,}053]$ (Section 4.1), and one public M covers all of them.

Since s begins with the constant 1, the update by $\pm X^j s$ changes coefficient j of z_0 by ± 1 and so flips its parity. For $\beta \in \{0,1\}^{256}$, let D_β be the target distribution on responses whose leading parities $z_0 \bmod 2$ equal β (Section 4.1); the mask y follows D_0. The restart probability $1-f-g$ of one update depends on the key and on z, but its average does not: if z follows D_β with $\beta_j = 0$, the update survives with probability exactly

$$\lambda = \frac{Z_{\text{odd}}}{M\, Z_{\text{even}}},$$

and the surviving response follows $D_{\beta \oplus e_j}$ (Lemma 4.2). Here Z_{even} and Z_{odd} are the total masses of the even and odd values of one coefficient of z_0. A challenge c of weight w therefore survives its updates with probability λ^w, which would reveal w. Following Gärtner, the signer appends $128 - w$ independent tests that each pass with probability λ; every challenge has $w \le 128$. Each attempt then survives with probability λ^{128} and outputs a sample of D_c, whatever the key and challenge (Theorem 4.3). For Pretzel's parameters $Z_{\text{odd}} \approx Z_{\text{even}}$, $\lambda \approx 1/M$ and $\lambda^{128} \approx M^{-128} \approx 0.86$, which is why M is taken as small as possible. Theorem 4.3 assumes a challenge independent of the mask, but in signing the challenge is computed from the commitment U of y. Let $\mathcal{Z}_U$ be the set of responses z for which Hz gives commitment U; the challenge is constant on $\mathcal{Z}_U$. Since $Hv = 0$ for every rotation v, we have $\mathbf{1}_{\mathcal{Z}_U}(z \pm v) = \mathbf{1}_{\mathcal{Z}_U}(z)$, so restricting to $\mathcal{Z}_U$ commutes with the updates and dummy tests. Appendix A.1 proves that, given its commitment U, a surviving response is distributed as D_c conditioned on $\mathcal{Z}_U$, where c is the challenge for U. This distribution depends on U and c but not on the secret key.

- **Cheap step decisions.** Each step must subtract, add, or restart with probabilities f, g and $1-f-g$. These are infinite series in θ, so comparing a random coin u with them directly needs high-precision arithmetic at every step. Pretzel avoids this in two ways. First, it compares exponents. For $a \ge 0$,

$$g = \frac{1}{M} \sum_{m \ge 1} (-1)^{m-1} \theta^{bm^2+2am},$$

and every exponent is an integer. The signer computes once per coin the integer ℓ with $\theta^\ell \le Mu < \theta^{\ell-1}$, so that $u < \theta^n/M$ exactly when $\ell > n$. Deciding whether to add then needs only integer comparisons, as in SHUTTLE [33, Section 3.3]. Second, the choice between subtracting and restarting involves a correction that the signer computes at low precision. In the rare cases too close to call, it subtracts provisionally and rechecks at high precision after all 128 steps, rejecting the attempt if the recheck fails. A surviving attempt has made the same decisions as the high-precision rule (Appendix D).

Pretzel's secret vector $s = (1, s_0, e)$ begins with the constant polynomial 1, so each update changes the first block of the response by ± 1 in a single coefficient, far less than it changes the other blocks. Pretzel therefore samples the first block from a distribution about seven times narrower than the second block's. SHUTTLE similarly narrows its first block, by scaling and rounding [33, Section 4.1], but Pretzel instead assigns each block its own width.

2.3 Signature format

For Pretzel-(4,4), the public key is $b = A_0 s_0 + e \bmod q$, where A_0 is a public 4×4 matrix of polynomials and s_0, e are short secret vectors. The secret has three parts, $s = (1, s_0, e)$. Writing $H = [-b \mid A_0 \mid I]$ gives

$Hs = 0$, so adding or subtracting a rotation of the secret leaves Hz unchanged modulo q.

The response starts as a random mask, then incorporates the challenge through updates using the secret. The signer samples a mask with the same structure,

$$y = (y_0, y_s, y_e),$$

and initializes $z = y$. For each position j with $c_j = 1$, a successful update changes all three parts together:

$$\begin{aligned} z_0 &\leftarrow z_0 \pm X^j, \\ z_s &\leftarrow z_s \pm X^j s_0, \\ z_e &\leftarrow z_e \pm X^j e. \end{aligned}$$

The sampler chooses the same sign for all three lines. Thus the response $z = (z_0, z_s, z_e)$ has three parts from the beginning. Here z_0 is one polynomial, while z_s and z_e each contain four polynomials.

The signature transmits z_0 and z_s, together with the salt and compressed commitment labels. Those labels let the verifier reconstruct an approximation to z_e, so that block can be omitted. The resulting signature size is

$$\underbrace{32}_{\text{salt}} + \underbrace{1160}_{\text{response}} + \underbrace{169}_{\text{labels}} = 1361 \text{ bytes}.$$

Pretzel's commitment labels play the role of the hints in ML-DSA and SHUTTLE. The commitment U records which of the 34 intervals contains each coefficient of $Hy + \delta$, where y is the initial mask and δ is a public offset. Since $Hz = Hy$, U also describes $Hz + \delta = -bz_0 + A_0 z_s + z_e + \delta$. The verifier can compute every term except z_e. It subtracts this computable part from the interval centers modulo q, using signed representatives, to reconstruct a residual r_e, which includes the rounding error. The signer checks the residual's coefficient cap before release; this puts the true label among at most four candidates near the verifier's estimate (Section 4.2). The signature spends one bit per label to choose between the two nearest candidates and lists the few exceptions separately, about 1.3 bits per label in all. The labels are cheap because the intervals are wide compared with r_e, so the verifier's estimate almost determines them. By our estimate, SHUTTLE's hint costs about 2.5 bits per coefficient, from its published rANS (range asymmetric numeral systems, an entropy-coding method [18]) model and frequency tables, which reproduce the stream size it reports [33, Table 7]. In Pretzel, the verifier combines the decoded head with r_e to form the reconstructed response ξ. It satisfies the exact commitment relation (10) used for extraction. The response coefficients are Rice-coded [40], so the many small coefficients take few bits. Signatures have a fixed length, and the signer restarts on the rare responses that do not fit.

Neither ML-DSA nor SHUTTLE adds an offset like δ. In Pretzel it is fixed for each key. Key generation derives it from a seed and does not test it on sample responses (Section 4.1).

One more difference from ML-DSA and SHUTTLE saves only a few bytes but may be of independent interest: Pretzel's signatures contain no challenge. Every coefficient of the initial mask's first block is even, and each update for a challenge bit $c_j = 1$ makes coefficient j of z_0 odd, so the verifier derives c from the parities of z_0. ML-DSA-65 and SHUTTLE-180 send a challenge of 48 bytes. Pretzel sends a salt of 32 bytes instead.

2.4 Quantum security

Classical proofs for Fiat–Shamir signatures obtain two responses to the same commitment and subtract them to get a short lattice vector. Doing this with a quantum forger is harder, because recording its first response can disturb the state needed to obtain the second. ML-DSA's and SHUTTLE's quantum analyses instead assume that self-target variants of Module-SIS are hard [26, 33].

Pretzel's proof avoids the self-target assumption in four steps.

Simulating signing. The ideal transport balance identity permits simulation without the secret: sample a response for a chosen challenge, recompute its commitment, and program the challenge hash to match. Appendix E bounds the difference between the complete real and simulated signing interactions, including failed attempts, retries, and later calls that reuse earlier randomness. The same comparison also accounts for dropping the requirement that keys have small norm. Fallahpour, Fehr, and Huang give a general version of this step for Fiat–Shamir with aborts, with a smaller loss than earlier reductions and without a stronger zero-knowledge assumption [19]; the appendix proves it directly for Pretzel's signer. Module-LWE then lets the reduction replace the public key b by a uniformly random vector, so the matrix $H = [-b \mid A_0 \mid I]$ is uniform apart from its identity block.

From a forgery to an interactive response. The signer derives the challenge by hashing the commitment, message digest, public key and salt. To extract a solution to Module-SIS, the proof needs to supply a fresh challenge after the commitment is fixed. The measure-and-reprogram technique of Don, Fehr, and Majenz [13] turns a quantum forger into a prover that answers such a challenge. This reduces its success probability by about a factor $(2q_B + 1)^2$, where q_B counts the forger's hash and verification queries (Section 5.3).

Measurement hiding. To extract, the proof must record the forger's response and then rewind the forger. For a quantum forger, recording is a measurement: it can change the forger's state, so after rewinding the forger might no longer succeed. The proof shows that, for accepted Pretzel responses and honest keys, no efficient forger can tell whether its response was measured unless it breaks Module-LWE (Section 5.2).

Once the key has been replaced by a uniform one, $H = [A_h \mid I]$ with A_h uniform, and every accepted response ξ solves $H\xi = y$ for a public target y. Multiplying each $|\xi\rangle$ by the phase $e^{2\pi i[v,\xi]_q/q}$ with v uniform removes all interference between distinct responses, which has the same effect as measuring them. The proof instead takes $v = (A_h^\dagger u - w,\ u)$ from an auxiliary Module-LWE sample $w = A_h^\dagger s + e$:

$$v = H^\dagger(u - s) + (-e,\ s).$$

The first term multiplies every solution of $H\xi = y$ by the same phase, which cannot be observed. Only the short vector $(-e, s)$ acts, and it removes interference only partially. A forger that detects measurement therefore distinguishes these phases from uniform ones, up to the loss caused by the partial effect. An interference experiment recovers the forger's gap divided by a factor κ that bounds this loss. The secant sum $\Lambda(\xi) = \sum_i 2t_i \log_2 \sec(2\pi\xi_i/q)$ is exactly the logarithm of the quantity that determines it. The verifier's secant check enforces $\Lambda(\xi) \le \log_2 \kappa$, which gives

$$\text{detection gap} \le \kappa\ \mathrm{Adv}_{\mathsf{MLWE},\mathrm{aux}} + \text{implementation errors}, \qquad \kappa \approx 2^{29.4}.$$

A filtering argument lowers the factor to $\sqrt{\kappa} \approx 2^{14.7}$ for the event the proof needs. Pretzel's narrow responses keep Λ small, so the check rarely rejects honest signatures. The argument adapts the LWE-phase technique of Poremba [38, full version, Section 5.1, Theorem 4] to Module-LWE, and it covers honestly generated keys only: for an adversarially chosen key, measurement can be detected (Remark B.3).

This avoids the larger parameters of earlier proofs. Poremba's argument and those of Liu and Zhandry [28], of Liu, Montgomery, and Zhandry [27] and of Jackson, Miller, and Wang [25] need the structured phases to leave the response state almost unchanged. That requires q to be much larger than the response times the noise, for example $q \approx 2^{43.5}$ in Jackson, Miller, and Wang [25, Section 4.3, Eq. (48)]. Kiltz, Lyubashevsky, and Schaffner instead use lossy keys, which also needs larger parameters [26]. Pretzel allows the phases to change the state substantially and pays the factor κ instead. The verifier checks the secant bound by summing entries from an integer table.

Extraction. Two accepted responses $\xi \neq \xi'$ for the same commitment give a Module-SIS solution: verification fixes $H\xi$ through the commitment, so $\eta = \xi - \xi'$ is short, nonzero, and satisfies $H\eta = 0 \pmod q$. Let p be the forger's success probability. The proof obtains the second response in one of two ways.

- *Strong unforgeability.* A new signature that reuses the digest, salt and commitment of a real signature already provides it, and subtracting the two reconstructed responses gives η (Section 5.1), as in Gärtner [22].
- *Ordinary unforgeability.* The proof records the forger's response, rewinds, and obtains a second accepted response for a different challenge. This succeeds with probability about p^3, before the factor κ for recording the response (Section 5.3).

If the forger's computation after the challenge is reversible and accepts challenges in superposition (Corollary F.6), ordinary unforgeability avoids rewinding. We notice that one response suffices: the proof tests whether reversing that computation restores the uniform superposition of challenges, following Unruh and Dall'Agnol and Spooner [8,43] as used by Jackson, Miller, and Wang [25]. The test succeeds with probability at least p^2 before recording, so the bound depends on p^2 instead of p^3.

3 Preliminaries

We analyze security against quantum polynomial-time (QPT) adversaries. Hash functions are modeled as random functions that an adversary can query in superposition; this is the *quantum random-oracle model* (QROM).

3.1 The ring and short polynomial vectors

All polynomial arithmetic uses

$$\mathcal{R} = \mathbb{Z}[X]/(X^{256} + 1), \qquad \mathcal{R}_q = \mathcal{R}/q\mathcal{R}, \qquad q = 7681.$$

We represent each element of $\mathcal{R}$ by its unique polynomial of degree less than 256. Vectors and matrices over $\mathcal{R}_q$ are arrays of these polynomials with coefficients reduced modulo q.

For $x \in \mathbb{Z}_q$, $\mathrm{center}_q(x)$ denotes the representative in $(-q/2, q/2]$. We apply this map coefficientwise to polynomials and vectors. For $x \in \mathcal{R}^k$, $\|x\|_2$ and $\|x\|_\infty$ are the Euclidean and maximum norms of the $256k$ integer coefficients in these representatives.

The centered binomial distribution CBD_t on $\mathbb{Z}$ is the sum of t independent differences of uniform bits. It is supported on $[-t, t]$, has mean zero and variance $t/2$. Applied to a polynomial or a vector, CBD_t samples every coefficient independently. Pretzel samples candidate secret and error coefficients from CBD_1 before applying the key-norm restriction of Section 4.1.

3.2 The two lattice problems

The security proof reduces forgery to the following two problems.

Module-LWE. For ranks k, n and secret and error distributions χ_s, χ_e on $\mathcal{R}$, a Module-LWE sample is $(A, As + e)$ with $A \leftarrow \mathcal{R}_q^{k \times n}$ uniform, $s \leftarrow \chi_s^n$, and $e \leftarrow \chi_e^k$, all independently sampled. The decisional problem is to distinguish it from (A, u) with $u \leftarrow \mathcal{R}_q^k$ uniform and independent of A. A distinguisher $\mathcal{D}$ has advantage

$$\mathrm{Adv}_{\mathsf{MLWE}}(\mathcal{D}) = \left|\Pr[\mathcal{D}(A, As + e) = 1] - \Pr[\mathcal{D}(A, u) = 1]\right|.$$

In the uniform-matrix model, Pretzel's core pair (A_0, b) is such a sample with $\chi_s = \chi_e = \mathrm{CBD}_1$ before the key-norm restriction of Section 4.1. The reduction removes this restriction in the comparison (F.1). The security proof also uses an auxiliary instance with CBD_2 secret and CBD_6 error (Section 5.2).

Normal-form Module-SIS. For a norm bound β and a uniform matrix $A_h \in \mathcal{R}_q^{m\times(\ell+1)}$, let

$$H = [A_h \mid I_m].$$

The problem is to find

$$d \in \mathcal{R}^{\ell+1+m} \setminus \{0\}, \qquad Hd = 0 \pmod q, \qquad \|d\|_2 \le \beta.$$

We write $\mathsf{MSIS}^{(2)}_{\mathcal{R},q,\ell,m,\beta}$ for this problem. A solver's success probability is the probability of outputting such a d, over the uniform choice of A_h and the solver's randomness and measurements. Requiring $\beta < q$ rules out the trivial solutions with a single coefficient equal to q.

In the construction, the extra column of A_h holds $-b$, alongside A_0. Section 5.1 explains how the reduction replaces A_h by a uniform matrix.

4 Pretzel Construction

4.1 Parameters and Key Generation

We specify Pretzel-(4,4), which targets category 3. Its public matrix A_0 is 4×4 over the polynomial ring, its public key has 1742 bytes, and its signatures have 1361 bytes.

Ring, key relation, and response weights. Work in $R_q = \mathbb{Z}_q[X]/(X^{256}+1)$ with $q = 7681$ and $(\ell, m) = (4, 4)$. Expand a 32-byte public matrix seed into $A_0 \in R_q^{4\times4}$ using the matrix domain specified in Appendix D. The proof models A_0 as uniform; the implementation expands the seed with SHAKE256. For a candidate key, sample $s_0 \leftarrow \mathrm{CBD}_1^4$ and $e \leftarrow \mathrm{CBD}_1^4$, put $b = A_0 s_0 + e \pmod q$, and define

$$A_h = [-b \mid A_0], \qquad H = [A_h \mid I_4], \qquad s = (1, s_0, e)^T.$$

Then

$$Hs = 0 \pmod q. \tag{1}$$

The response $z = (z_0, z_s, z_e)$ has integer coefficients and one block for each part of s. To give each block its own width, we define the weighted sum of squared integer coefficients

$$E(z) = 9216\|z_0\|_2^2 + 196\|z_s\|_2^2 + 121\|z_e\|_2^2. \tag{2}$$

For each challenge c defined below, the ideal target distribution D_c assigns mass proportional to $\theta^{E(z)}$ to responses whose leading parity is c. Here $0 < \theta < 1$ is a fixed rational number whose denominator is a power of two (Appendix D). Section 4.3 defines the normalization and gives convergent rational bounds that enclose the ideal step probabilities to any precision.

Bounded key generation. Key generation samples a matrix seed and a private randomness key K_{rand}, then tries at most 512 candidates for (s_0, e). A candidate is accepted when

$$120{,}000 \le 23{,}716 + 196\|s_0\|_2^2 + 121\|e\|_2^2 \le 244{,}553. \tag{3}$$

Since $E(s) = 9216 + 196\|s_0\|_2^2 + 121\|e\|_2^2$, condition (3) says that $E(s) \in [105{,}500, 230{,}053]$. Section 4.3.2 uses this range to choose one public scaling constant that works for every key. If all 512 candidates fail, key generation stops without returning a key pair.

The public offset and key contents. The public key also fixes an offset $\delta \in R_q^4$, called the *dither*, which is added to Hy before the commitment is rounded (Section 4.2). To keep the public key short, the key stores a 32-byte seed, and δ is derived from it by hashing. The derivation is rejection sampling with a limit on the number of hash output words it reads, so its running time is bounded; it fails when too few usable values appear, which happens with probability below 10^{-110} for each seed. Key generation therefore tries up to 2048 seeds in a fixed order and keeps the first that works; if none does, it fails and the rest of the key is not replaced. It does not test the resulting dither: the choice of dither affects how often signing retries, and Section 4.2 bounds that effect for every dither. Appendix D.2 gives the exact procedure for Pretzel-(4,4), Appendix I.1 for Pretzel-(5,5) and Section H.1 for Pretzel-(3,3).

Write $\rho_A, \rho_\delta \in \{0,1\}^{256}$ for the matrix and dither seeds. Set $\widehat{b} = \mathrm{NTT}(b)$; $K_{\mathrm{rand}} \in \{0,1\}^{256}$ is the private randomness key above. All procedures in this section use the fixed parameters and identifying label of Pretzel-(4,4). Leaving that label implicit, we write the keys as

$$\begin{aligned} \mathsf{pk} &= (\rho_A, \widehat{b}, \rho_\delta), \\ \mathsf{sk} &= (\rho_A, s_0, e, K_{\mathrm{rand}}, h_{\mathsf{pk}}, \rho_\delta). \end{aligned}$$

Let Encode denote the byte encoding and $\mathrm{Expand}_A, \mathrm{Expand}_\delta$ the deterministic SHAKE256 procedures of Appendix D. Writing $\mathrm{SHAKE256}_{32}$ for SHAKE256 with 32 output bytes,

$$\begin{aligned} h_{\mathsf{pk}} &= \mathrm{SHAKE256}_{32}\Big(\mathrm{Encode}(\rho_A, \widehat{b}, \rho_\delta)\Big), \\ A_0 &= \mathrm{Expand}_A(\rho_A), \\ \delta &= \mathrm{Expand}_\delta(\rho_A, \widehat{b}, \rho_\delta). \end{aligned}$$

The challenge set. The hash challenge c is encoded in the 256 coefficients of the response polynomial z_0, with an even coefficient representing zero and an odd coefficient representing one. We choose uniformly from 2^{255} possible challenges, each containing at most 128 ones. This limits each signing attempt to at most 128 secret-key updates. The challenge set is

$$C_{255} = \{c \in \{0,1\}^{256} : \mathrm{wt}(c) < 128\} \cup \{c \in \{0,1\}^{256} : \mathrm{wt}(c) = 128,\ c_{255} = 0\}. \tag{4}$$

Here $\mathrm{wt}(c)$ denotes the Hamming weight of c, the number of one bits. The set C_{255} contains 2^{255} vectors, each of weight at most 128. To map a 255-bit string u into the challenge set, append a zero bit and call the result $x = (u, 0)$. Output x if $\mathrm{wt}(x) \le 128$, and otherwise output its bitwise complement. The last bit records whether we flipped the bits.

We write the even/odd encoding of c in z_0 using the *parity* function. For $z_0 = \sum_{i<256} z_{0,i} X^i$ with integer coefficients write $\mathrm{parity}(z_0) = (z_{0,i} \bmod 2)_{i<256} \in \{0,1\}^{256}$, and require

$$c = \mathrm{parity}(z_0). \tag{5}$$

4.2 Commitment and Response Reconstruction

To reduce signature size, the signer transmits $z_h = (z_0, z_s)$ and omits z_e. This leaves a missing term in

$$Hz = \underbrace{A_h z_h}_{\text{known to the verifier}} + \underbrace{z_e}_{\text{omitted}}.$$

The verifier needs a small response consistent with the commitment, in the sense of the relation (10) below. It does not need the original z_e. We first explain how interval labels allow reconstruction, then how the transmitted response helps compress those labels.

Intervals, labels, and centers. For a random mask y, form $t = Hy + \delta \pmod q$, using the public dither fixed by the key. Since H has m rows, t consists of m polynomials, each with 256 coefficients, for $256m$ coefficients in total ($m = 4$ and 1024 coefficients for Pretzel-(4,4)). Represent each coefficient by an integer in $\{0, \dots, q-1\}$. Partition these values into L intervals of consecutive integers. Their endpoints and labels are

$$a_j = \left\lceil \frac{qj}{L} \right\rceil \quad (0 \le j \le L), \qquad I_j = \{a_j, \dots, a_{j+1} - 1\} \quad (0 \le j < L).$$

For $x \in \{0, \dots, q-1\}$, the function

$$Q_{q,L}(x) = \left\lfloor \frac{Lx}{q} \right\rfloor \in \{0, \dots, L-1\} \tag{6}$$

returns the label j of the interval containing x. A label identifies the interval, but loses the exact position within it. To approximate the coefficient, choose an integer center $C(j) \in I_j$ nearest to the midpoint

$$m_j = \frac{a_j + a_{j+1} - 1}{2}.$$

When two integers are equally close, which happens exactly for the intervals with an even number of integers (half-integer midpoint), alternate between the lower and upper choices across those intervals in increasing order of j, starting with the lower. We write $Q = Q_{q,L}$ below. Every argument of Q is first reduced modulo q to its representative in $\{0, \dots, q-1\}$. Both Q and C act on vectors one coefficient at a time.

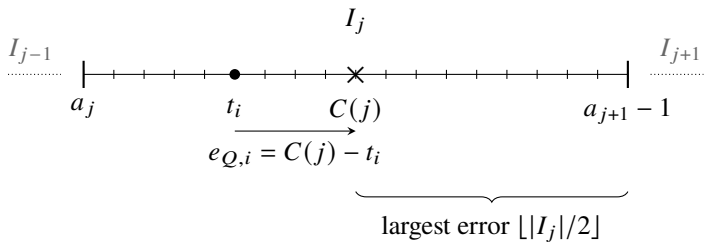
The commitment and its rounding error. The commitment consists of the interval labels of all coefficients of t:

$$U = Q(Hy + \delta). \tag{7}$$

The sampler changes y into the response z by adding or subtracting $X^j s$. Since $Hs = 0$, these updates preserve $Hy = Hz$ modulo q (Section 2.1), so

$$U = Q(Hz + \delta). \tag{8}$$

Replacing each coefficient by the center of its interval gives $[C(U)]_i = C(U_i)$ and rounding error $e_Q = C(U) - t$, using representatives of t in $\{0, \dots, q-1\}$. The diagram shows one coefficient t_i in interval I_j, $j = Q(t_i)$, its center ($\times$) and the signed rounding error.



The largest rounding error in an interval occurs at an endpoint and equals $\lfloor |I_j|/2 \rfloor$. The largest interval contains $\lceil q/L \rceil$ integers, so

$$\max_{0 \le j < L} \max_{u \in I_j} |C(j) - u| = \left\lfloor \frac{\lceil q/L \rceil}{2} \right\rfloor .$$

Appendix D.5.1 gives the exact error probabilities and moments for a uniform input.

Reconstructing the omitted block. The verifier computes $x = A_h z_h + \delta$ from the public key and transmitted response. Since $t_i \equiv x_i + z_{e,i} \pmod q$, subtracting x_i from the interval center approximates the missing coefficient. For the whole block, the verifier reconstructs

$$r_e = \mathrm{center}_q\big(C(U) - x\big) = \mathrm{center}_q(z_e + e_Q), \tag{9}$$

where center_q selects the signed representative modulo q. The block r_e is also called the residual. The reconstructed response $\xi = (z_h, r_e)$ satisfies the exact relation

$$H\xi = C(U) - \delta \pmod q. \tag{10}$$

The bounds in Section 4.4 apply to this reconstructed response. The signer computes the same response and retries if a check fails. These checks apply to r_e itself and do not assume uniform or independent rounding errors.

Compressing the labels. The transmitted response also helps the verifier recover the labels from a shorter encoding. Write B for the bound on $|r_{e,i}|$. A label j is possible only if its reconstructed coefficient satisfies

$$\big|\mathrm{center}_q(C(j) - x_i)\big| \le B. \tag{11}$$

Signing enforces $\|r_e\|_\infty \le B$ and retries the attempt otherwise (Section 4.4). Since $\mathrm{center}_q(C(U_i) - x_i) = r_{e,i}$ by (9), the true label U_i of every signature satisfies (11), so the verifier never excludes it. Thus the verifier can exclude centers more than B away from x_i, measuring distance modulo q. The signature need only distinguish the remaining labels. Narrower intervals reduce rounding errors, but put centers closer together, leaving more labels within the same bound.

For Pretzel-(4,4), we choose $L = 34$ and $B = 450$, which gives maximum rounding error 113. For every x_i, at most four labels satisfy (11) (Lemma D.2). Section 4.5 specifies an ordered four-entry recovery list that contains every label satisfying (11) (Lemma D.1).

Choosing the dither. The rounding error depends on δ, since shifting t moves each coefficient within its interval. By (9) this error adds to z_e in r_e, so it uses up part of the bound B, and a larger error means more signing retries. Because the dither is fixed in the key, this effect applies to every signature under that key. Key generation does not test the dither (Section 4.1), so we bound the effect for every dither instead. The intuition is that the public matrix spreads the response out. When no small combination of two rows of A_0 has too many coefficients of the same size (the spread condition of Lemma D.3), each value that is rounded, and each pair of them, is close to uniform whatever the dither, so the rounding error behaves as it would for a random offset. For such a matrix and every fixed dither and challenge, an attempt with fresh randomness fails the public checks with probability below 0.293. A uniformly random matrix violates the spread condition with probability below $1.423 \cdot 10^{-37}$. These bounds concern retries only: the security proof uses the same dither selection in every step and does not rely on them (Theorem 5.6). Appendices I.1 and H.1 treat Pretzel-(5,5) and Pretzel-(3,3).

4.3 The Iterative Transport Sampler

The challenge is encoded in the parities of z_0. Each step flips one parity bit by adding or subtracting $X^j s$, with probabilities chosen so that the output distribution depends only on the challenge, not on s. The signer starts with a random mask y from D_0, defined below, whose first polynomial y_0 has only even coefficients. Each step adds or subtracts $X^j s$ from the response z, or restarts signing. We call this randomized update a *transport step*: it changes one coefficient of z_0 from even to odd while preserving Hz, and hence the commitment U. For a challenge fixed independently of this initial sample, the surviving response has the corresponding target distribution. When the challenge depends on the commitment, the balance identity (22) holds separately for each commitment value (Appendix A.1).

4.3.1 The target distribution

Let θ be the exact dyadic base fixed in Appendix D. We assign each response the unnormalized weight

$$p(z) = \theta^{E(z)}, \tag{12}$$

with E from (2). The weights factor over coordinates. The rational base permits exact enclosing intervals for each of the probabilities f and g below; infinite normalizers and alternating series are handled by directed bounds. For every parity vector $\beta \in \{0,1\}^{256}$, let D_β be the normalized restriction of p to $z \in \mathcal{R}^{1+\ell+m}$ with integer coefficients and

$$\text{parity}(z_0) = \beta. \tag{13}$$

The target response distribution for challenge $c \in C_{255}$ is D_c. Under D_β the coordinates are independent. Coefficient j of z_0 is restricted to even values when $\beta_j = 0$ and odd values when $\beta_j = 1$; the coefficients of z_s and z_e have no even/odd restriction.

4.3.2 One transport step

Suppose the step must change parity bit j, and define $v = X^j s$. The leading component of v is X^j, so both $z - v$ and $z + v$ flip exactly parity bit j. Rotation only permutes coefficients and changes signs, so $E(v) = E(s)$, which we write b throughout this section. Let $\langle x, y\rangle_W = 9216\langle x_0, y_0\rangle + 196\langle x_s, y_s\rangle + 121\langle x_e, y_e\rangle$, the bilinear form with $E(x) = \langle x, x\rangle_W$; its coefficients 9216, 196 and 121 are the coordinate weights of E in (2), and let

$$a = \langle z, v\rangle_W. \tag{14}$$

Then $E(z + kv) = E(z) + 2ka + k^2 b$ for every integer k.

The probabilities are chosen by looking backward from an output x. It can be reached from $x + v$ by subtraction or from $x - v$ by addition. The contributions from these two inputs must sum to $p(x)/M$, where $M \geq 1$ is one public scaling factor for every x and every key. This is the balance identity (22). Along a line of responses differing by multiples of v, the weight ratios are powers of θ determined by a and b. Gärtner's alternating series solves this balance equation.

For $u \geq 0$, define the convergent alternating series

$$T(u, b) = \sum_{k=0}^{\infty} (-1)^k \theta^{2ku + k^2 b}. \tag{15}$$

The numbers $\widehat{f} = Mf$ and $\widehat{g} = Mg$, before division by M in (17) and (18), sum to exactly one when $a \geq b$ or $a < -b$. For $-b \leq a < b$ their sum can exceed one, up to the exact supremum $\mathcal{M}(b)$ of (20), which grows with the weighted squared norm $b = E(s)$ (Lemma 4.1). Both numbers are therefore divided by one public

constant $M \geq \mathcal{M}(b)$, valid for every key the signer may use, so that the average probability of surviving a step does not depend on the key. Gärtner divides by a majorant from Poisson summation of this sum [22]. We instead divide by one public rational constant M with denominator 2^{16}, certified to lie above the exact supremum $\mathcal{M}(b)$ for every key that passes the key check (3). The fixed rational envelope is

$$M = \frac{65{,}612}{65{,}536}. \tag{16}$$

The probabilities of moving to $z - v$ and $z + v$ are, respectively,

$$f(a,b) = \begin{cases} T(a,b)/M, & a \geq b, \\ \theta^{b-2a} T(b-a,b)/M, & a < b, \end{cases} \tag{17}$$

$$g(a,b) = \begin{cases} \theta^{b+2a} T(b+a,b)/M, & a \geq -b, \\ T(-a,b)/M, & a < -b. \end{cases} \tag{18}$$

The signer outputs $z - v$ with probability f and $z + v$ with probability g. Otherwise the step aborts, and an abort restarts the whole signing attempt with a fresh salt and mask (Section 4.4).

Relation to Gärtner's sampler. The series (15) and the probabilities of subtracting or adding v in (17)–(18) are Gärtner's [22], applied to the weighted squared norm (2), while the exact envelope (20), its rational rounding (16), and the certified bounds that define the sampler are new.

Lemma 4.1 (The probabilities of subtracting or adding v are valid)**.** Let $\widehat{f} = Mf$ and $\widehat{g} = Mg$ be the numerators before division by M in (17)–(18). For every $b > 0$ and $a \in \mathbb{Z}$,

$$\widehat{f}(a,b) \geq 0, \qquad \widehat{g}(a,b) \geq 0, \qquad \widehat{f}(a,b) + \widehat{g}(a,b) \leq \mathcal{M}(b), \tag{19}$$

where

$$\mathcal{M}(b) = 1 + \frac{\vartheta_4(0,\theta^b)}{\theta^b}, \qquad \vartheta_4(0,x) = \sum_{k \in \mathbb{Z}} (-1)^k x^{k^2}. \tag{20}$$

The exact envelope $\mathcal{M}$ is increasing. At the maximal allowed weighted squared norm $E(s)$, the finite evaluation in Appendix A gives $\mathcal{M}(230{,}053) < 65{,}612/65{,}536 = M$. Hence every key accepted by (3) satisfies

$$0 \leq f(a,b), \qquad 0 \leq g(a,b), \qquad f(a,b) + g(a,b) \leq 1, \tag{21}$$

so f and g are valid probabilities of moving to $z - v$ and $z + v$, respectively, and $1 - f - g$ is a valid restart probability.

The proof is in Appendix A.

4.3.3 Exact sampling of a step

The three-outcome step can be defined using one independent uniform $U \in [0, 1)$: choose minus for $U < f$, plus for $f \leq U < f + g$, and abort otherwise. This uses the two thresholds f and $f + g$, rather than two independent Bernoulli tests. It admits an exact random-bit implementation. The terms of (15) decrease to zero because $u \geq 0$ and $b > 0$; consecutive even and odd partial sums enclose $T(u,b)$, with gap bounded by the next term. The rational prefactors in (17)–(18) preserve convergent rational enclosures for both thresholds.

Refine these enclosures while revealing successively more binary digits of U, until its enclosing dyadic interval lies wholly in one of the three outcome regions. The thresholds' enclosures and the interval for U shrink to points. Therefore the decision terminates unless U equals one of the thresholds, an event of probability zero, and gives exactly the probabilities f, g, and $1 - f - g$.

4.3.4 The balance identity for one step

Lemma 4.2 (Weighted transport balance, after Gärtner)**.** For every key, every shift $v = X^j s$ with $b = E(v)$ and every x with integer coefficients,

$$p(x+v)\, f(\langle x+v, v\rangle_W, b) + p(x-v)\, g(\langle x-v, v\rangle_W, b) = \frac{p(x)}{M}. \tag{22}$$

Therefore one step changes the selected parity bit and produces the target distribution for the new parity pattern. The total surviving mass changes by a constant that does not depend on the output.

The proof is in Appendix A.

4.3.5 The final response distribution

Let

$$Z_{\text{even}} = \sum_{x \text{ even}} \theta^{9216x^2}, \qquad Z_{\text{odd}} = \sum_{x \text{ odd}} \theta^{9216x^2},$$

the even and odd masses of one leading coordinate, whose weight in E is 9216. Their ratio determines how much probability mass survives a step from even to odd parity. Define

$$\lambda = \frac{Z_{\text{odd}}}{M Z_{\text{even}}}. \tag{23}$$

To make the survival probability independent of the challenge weight, the signer follows the transport steps of a weight-w challenge by $128 - w$ independent Bernoulli(λ) tests and restarts if any fails. Every challenge in C_{255} has weight $w \le 128$ by (4) (Section 4.1), so $128 - w \ge 0$. This padding is Gärtner's device (Section 4.1 of [22]). We call the padded tests dummy tests.

Theorem 4.3 (Distribution after transport)**.** Assume the key satisfies (3). Let K_v be the substochastic kernel of one step with shift v. It maps z to $z-v$ with probability $f(\langle z, v\rangle_W, b)$, to $z+v$ with probability $g(\langle z, v\rangle_W, b)$, and to nothing otherwise. Let $c' \in \{0,1\}^{256}$ be the current leading parity vector with $c'_j = 0$, and let e_j be the jth unit vector. Including the chance of restart, one step at position j gives

$$D_{c'} K_{X^j s} = \lambda D_{c' \oplus e_j}. \tag{24}$$

For a challenge c of weight w fixed independently of the initial D_0 sample, process its nonzero positions in any fixed public order. The unnormalized measure of the surviving output is then

$$\lambda^w D_c. \tag{25}$$

After the $128 - w$ dummy tests the unnormalized surviving measure is

$$\lambda^{128} D_c. \tag{26}$$

The surviving mass λ^{128} is independent of both the challenge and the secret key, and the distribution D_c depends only on c.

Proof. Appendix A.1. □

When the challenge depends on the commitment, the same identity holds separately for each commitment value, because every update preserves Hz. Appendix A.1 derives the resulting response law, and Appendix E compares the implementation with it.

4.4 Signing and Verification

The numerical bounds and local-label encoding below specify Pretzel-(4,4). The Pretzel-(3,3) specialization, including its challenge representation, is given in Appendix H. Pretzel-(5,5) uses the parameters and signature encodings of Section I.1, with the same transport and commitment-recovery steps.

The key fixes A_h and the public dither δ. Hash inputs use separate domains and length fields that bind the parameter-set and encoding identifier, public key, application context, and message. The challenge hashes the commitment labels themselves, not their compressed form, which depends on the response and so does not exist when the challenge is computed. Exact hash input encodings are specified in Appendix D.

Signing. Compute the message digest μ once. For each of at most 512 attempts, request fresh random bytes and derive the salt and the random bits used by the sampler, with separate labels for their different uses. If the source of random bytes fails, return an error immediately. An ordinary rejection advances to the next attempt.

1. Run the finite proposal sampler of Appendix D, approximating D_0, and obtain the proposal y and the 32-byte salt ρ. Record any sampler failure for rejection at the end of the attempt.

2. Form $t = Hy + \delta$ and commitment labels $U = Q_{q,34}(t)$. Hash the encoded public key, message digest μ, salt ρ, and labels U, and apply Fold_{255} to obtain $c \in C_{255}$.

3. Start at $z = y$. In increasing order of $j \in \mathrm{supp}(c)$, apply the transport update with $v = X^j s$, using the finite implementation in Appendix D of the probabilities in Section 4.3. Apply the remaining dummy tests so that the attempt has 128 tests in total, counting both secret-key updates and dummy tests. Record any transport, dummy-test, or deferred-check failure. Process all scheduled positions and complete all final checks before rejecting a failed attempt.

4. Split $z = (z_h, z_e)$. Compute $x = A_h z_h + \delta$ in R_q^4; transport gives $x \equiv t - z_e \pmod q$. Set $r_e = \mathrm{center}_q(C(U) - x)$. Check that $\xi = (z_h, r_e)$ passes the public response checks of Table 4.

5. Only if all checks pass, encode z_h and the local indices of U, and return the salt and those two fields. If encoding nevertheless fails, reject the attempt.

If all 512 attempts reject, signing fails. Pretzel-(4,4) needs about 1.21 attempts per signature on average (Section 2.2). The analysis models the transport decisions with independent random bits. The implementation's bounded precision, and its reuse of low-order random bits across steps, are accounted for in Appendix E.

Verification. Verification performs the following checks.[2]

1. Require the exact signature length. Parse the salt and canonically decode $z_h = (z_0, z_s)$ and the residual-label index fields. Reject invalid counts, ordering, padding, or coefficient encodings.

2. Compute $x = A_h z_h + \delta$. For each coordinate, recover its commitment label U_i from the decoded local index and public context x_i. Reconstruct $r_e = \mathrm{center}_q(C(U) - x)$.

3. Require that the reconstructed response $\xi = (z_h, r_e)$ passes the public response checks of Table 4.

4. Recover $c = \mathrm{parity}(z_0)$ and require $c \in C_{255}$. Recompute the hash with the specified input encoding from the message, public key, salt, and recovered labels. Accept exactly when its folded digest equals c.

Table 4: Public response checks for Pretzel-(4,4), applied by the signer before release and by the verifier to the same reconstructed response $\xi = (z_h, r_e)$.

Check	Bound
Coefficients	$\|z_0\|_\infty \le 288$, $\|z_s\|_\infty \le 255$, $\|r_e\|_\infty \le 450$
Euclidean norms	$\|z_h\|_2 \le 1593$, $\|r_e\|_2 \le 3068$
Secant sum	$\sum_i \lceil w_i 2^{20} \log_2 \sec(2\pi\xi_i/q) \rceil \le 30{,}850{,}912$, with $w_i = 12$ on z_h and $w_i = 4$ on r_e
Fits in the encoding	quotient sum at most 1088; at most 50 local-label exceptions

Public information. The public key determines A_h and the dither δ, which is expanded from its seed. Both parties compute the message digest μ.

Signer: finish an attempt

1. Start with salt ρ, labels U, and response $z = (z_h, z_e)$ from iterative signing, where $U = Q(Hy + \delta)$ and $Hz = Hy \pmod q$.
2. Compute $x = A_h z_h + \delta$ and $r_e = \mathrm{center}_q(C(U) - x)$.
3. Check the response and encoding bounds below. Discard the attempt if a check fails.
4. Encode z_h and the compressed labels, and return ρ and these two encodings. Discard the attempt if encoding fails.

Verifier

1. Decode the signature. Reject an invalid length or encoding.
2. Compute $x = A_h z_h + \delta$. Recover U from the compressed labels using x, then compute $r_e = \mathrm{center}_q(C(U) - x)$.
3. Check the same response and encoding bounds. Reject if any check fails.
4. Set $c = \mathrm{parity}(z_0)$. Accept only if $c \in C_{255}$ and
$$c = \mathrm{Fold}_{255}\big(\mathcal{H}(\mathsf{pk}, \mu, \rho, U)\big).$$

Signature: salt ρ, encoded $z_h = (z_0, z_s)$, and compressed labels U. The blocks z_e and r_e are not transmitted.

Shared response and encoding checks. For $\xi = (z_h, r_e)$, require

$$\underbrace{\|z_0\|_\infty \le 288, \quad \|z_s\|_\infty \le 255, \quad \|r_e\|_\infty \le 450,}_{\text{limit each coefficient}} \qquad \underbrace{\|z_h\|_2 \le 1593, \quad \|r_e\|_2 \le 3068}_{\text{limit the combined size of each block}}.$$

Also check the weighted secant sum required by the quantum proof and that the response fits in the encoding (Section 4.5).

Figure 2: Final signing checks and verification for Pretzel-(4,4). The signer checks the reconstructed response before releasing a signature. The preceding procedures specify the sampling steps, limit on signing attempts, and exact hash inputs. The hash input encoding is omitted.

The secant check uses upward-rounded table entries to bound the weighted log-secant sum. Its cap gives a uniform bound on the loss factor κ of measurement hiding (Section 2.4). Appendix B.2 gives the weights and the resulting value of κ. The implementation evaluates this check using the exact integer tables. Appendix B.2 gives their construction and the resulting measurement bound. Canonical coefficient and local-label coding and the checks that the response fits in the encoding are specified in Section 4.5.

Figure 2 summarizes the final signing checks and verification. Both parties reconstruct the same response and apply the same bounds.

Signing fails only if all 512 attempts are rejected. In the ideal sampler, an attempt with fresh randomness

[2] A prepared public key has already been unpacked from its stored form, with its matrix and offset expanded and validated. Verification can reuse these values across signatures; otherwise, in fresh verification, it computes and validates them first. Both cases then perform the same checks.

passes the transport steps and dummy tests with probability λ^{128} (Theorem 4.3), and its response then passes the public checks with probability above $1 - 0.293$ whenever the public matrix satisfies the spread condition of Lemma D.3. Each such attempt therefore succeeds with probability above $p_4 = \lambda^{128}(1 - 0.293) > 0.6$, and 512 of them all fail with probability below $(1 - p_4)^{512} < 2^{-694}$. Appendix D.5.4 extends this bound to the implemented signer. Correctness, in contrast, holds for every returned signature and every public matrix.

Theorem 4.4 (Correctness)**.** Under the specified arithmetic, hash input encoding, and encoding rules, every returned signature is accepted by verification for the same key, context, and message.

Proof. The proposal has even leading coefficients. Each transport update adds or subtracts $X^j s$, whose leading component is X^j, so the updates at exactly the positions in $\mathrm{supp}(c)$ give $\mathrm{parity}(z_0) = c$. Since $Hs = 0 \pmod q$, they also preserve $Hz = Hy \pmod q$, and hence the commitment U. The signer checks the reconstructed response and that it fits in the encoding before release. Lemma 4.6 guarantees that canonical decoding of the returned signature recovers the same head, labels, and residual. The verifier therefore applies the same public checks and recomputes the same challenge hash with the specified input encoding for the same key, context, message, and salt. Its folded digest equals $\mathrm{parity}(z_0) = c$. □

4.4.1 Verification with the challenge given instead of hashed

Definition 4.5 (Verification with a supplied challenge)**.** To separate response validity from the hash comparison, replace the latter by a prescribed $\chi \in C_{255}$ and require $\mathrm{parity}(z_0) = \chi$, and the parsed salt and recovered labels to equal ρ and U. This defines $V_{\mathrm{ID}}(\mathsf{pk}, \mu, \rho, U, \chi; Z)$ on the complete canonical signature byte string Z.

Every response accepted by verification with a supplied challenge satisfies the response conditions used by Theorem 5.1 and Theorem 5.4. Extraction also needs a second accepted response for the same first message, and measurement hiding a uniform public matrix, which Appendix B.5 obtains using Module-LWE. The security proof accounts for aborted attempts and retries, and for the fact that the dither is fixed at key generation rather than chosen afresh for each attempt.

4.5 Canonical Signature Encoding

The Pretzel-(4,4) signature consists of a 32-byte salt, a 1160-byte response head, and a 169-byte local-label field, for a total of 1361 bytes. These fields are concatenated in that order. Within each packed field, bit position t is bit $t \bmod 8$ of byte $\lfloor t/8 \rfloor$, with bit zero the least significant bit. Fixed-width integers are written least significant bit first. Coefficients are ordered by polynomial, then by degree from zero through 255, with z_0 before the four polynomials of z_s; labels follow the same polynomial and degree order. The hash consumes the recovered commitment labels as 1024 individual bytes. The challenge c is recovered by reading each coefficient of z_0 as even (zero) or odd (one). The omitted vector z_e is replaced by the reconstructed residual $r_e = \mathrm{center}_q(C(U) - A_h z_h - \delta)$ of (9), computed from $z_h = (z_0, z_s)$ and U; in general $r_e \neq z_e$. Neither c nor z_e requires a separate transmitted field.

Encoding of z_h. Map a signed coefficient to $v(z) = 2z$ for $z \geq 0$ and $v(z) = -2z - 1$ otherwise. For remainder width k, set $a = \lfloor v(z)/2^k \rfloor$ and $r = v(z) \bmod 2^k$, so $v(z) = 2^k a + r$ with $0 \leq r < 2^k$. The 256 leading coefficients use three remainder bits each and the 1024 remaining head coefficients use six. Store the leading remainders first, then the remaining head remainders, in coefficient order. They occupy $96 + 768 = 864$ bytes, followed by the quotient field. This is Rice coding [40].

In coefficient order, write each quotient a as a zero bits followed by a one. Require the sum of the 1280 quotients to be at most 1088. The quotient field occupies $(1280+1088)/8 = 296$ bytes, with unused bits after

its 1280 delimiters set to zero. Decoding rejects missing delimiters, quotient overflow, nonzero padding, and coefficients outside the respective caps 288 and 255. The encoding of z_h occupies $864 + 296 = 1160$ bytes.

Local commitment labels. Encoding each of the 1024 labels independently at fixed width requires six bits, since there are 34 possible labels. But the verifier can compute $x = A_h z_h + \delta$ from the head, and the true label of coordinate i lies close to $Q(x_i)$ because the residual is short. The signer therefore sends each label's position in a short list of candidates near $Q(x_i)$. For each public context $x_i = (A_h z_h + \delta)_i$, let $j = Q(x_i)$ and choose $s \in \{-1, 1\}$ toward the nearer adjacent quantizer center, using centered distances modulo q and labels modulo 34, with ties toward -1. Every label satisfying residual cap 450 occurs in

$$(j, j + s, j - s, j + 2s) \pmod{34}.$$

Lemma D.1 certifies this containment for every context. The fourth entry is checked against the residual cap. Write the local index as $b + 2e$, with $b, e \in \{0, 1\}$; usually $e = 0$, so the label is one of the two nearest candidates. Store the 1024 low bits b in 128 bytes. At most 50 positions, the exceptions, may have $e = 1$. The local-label field concatenates these low bits, the exception count, the exception low fields, and the bit vector specifying the high parts of those positions, in that order.

Encode the increasing exception positions a_i using Elias–Fano coding [44]. Store their count in one byte, their low four bits in 50 reserved four-bit fields in increasing exception order, and set bit $\lfloor a_i/16 \rfloor + i$ in a bit vector, with indices i starting at zero. The bit vector has $63 + 50 = 113$ bits, stored in 15 bytes. Let n be the stored count. The decoder requires $n \le 50$ and exactly n one bits in the bit vector. From their increasing positions h_i and the stored low four bits ℓ_i, it recovers $a_i = 16(h_i - i) + \ell_i$, with i starting at zero. It checks that the positions a_i are strictly increasing and lie between 0 and 1023. Unused low fields, unused high bits, and the seven final padding bits must be zero. The local-label field occupies $128 + 1 + 25 + 15 = 169$ bytes.

Because signatures have a fixed length, the signer rejects a response whose encoding would not fit. It runs these checks before releasing anything. Under the ideal response distribution, they reject at most a 0.000220 fraction of responses, for every public key, dither, and challenge (Section D.5).

Lemma 4.6 (Canonical bytes)**.** For either the Pretzel-(4,4) 1361-byte encoding or the Pretzel-(5,5) 1908-byte encoding, fix the public key, dither, salt ρ and commitment U. Let $\mathcal{R}_U$ consist of the integer responses $\xi = (z_h, r_e)$ with $H\xi = C(U) - \delta \pmod{q}$ that pass the public coefficient, Euclidean and secant checks of the chosen parameter set and encoding and the checks on encoding ranges and whether the response fits in the encoding. Let $\mathcal{S}_U$ consist of the strings that decode canonically with salt ρ and recovered commitment U to a response passing these same checks; no hash comparison is involved. Using the recovery lists of Lemma D.1, canonical encoding and decoding are inverse bijections between $\mathcal{R}_U$ and $\mathcal{S}_U$. The bijection restricts to any subset defined by a public predicate of ξ, in particular $\mathrm{parity}(z_0) \in C_{255}$ or $\mathrm{parity}(z_0) = \chi$ for a supplied challenge χ. Remark B.2 explains why the decoder can be run reversibly.

Proof. For the Pretzel-(4,4) 1361-byte encoding of Section 4.5 or the Pretzel-(5,5) 1908-byte encoding of Section I.1, fix $\mathsf{pk}, \delta, \rho, U$. Write E_U and D_U for the canonical encoder and decoder with these fixed controls. Here a response satisfying the public checks is an element of $\mathcal{R}_U$ and an accepted string an element of $\mathcal{S}_U$. The inverse identities are $D_U(E_U(\xi)) = \xi$ for every $\xi \in \mathcal{R}_U$ and $E_U(D_U(s)) = s$ for every $s \in \mathcal{S}_U$. The encoder emits the fixed salt ρ, including all 32 salt bytes; a string carrying another salt is rejected. For Pretzel-(4,4), the signed map, fixed remainder fields and exact delimiter count determine a unique head; the count, ordered positions and zero unused fields determine a unique local-label exception set. Distinct local candidates give a unique index for each recovered label. The residual and head determine the label center, which determines its label uniquely. The integer coefficient ranges exclude modular aliases.

The Pretzel-(5,5) 1908-byte case is the same argument, given in Appendix I.1. Lemma D.1 gives recovery-list containment, so every label satisfying the residual bound has such an index. □

Remark 4.7 (What the measurement application uses). The measurement application uses only injectivity and $E_U(D_U(s)) = s$ on accepted strings; those properties do not require recovery-list containment.

5 Security Analysis

5.1 Extracting a Short Kernel Vector

Any two distinct accepted responses for one first message yield a short kernel vector, whether or not their challenges coincide. This is the algebraic step in our reduction to normal-form Module-SIS. Obtaining both responses from a quantum forger requires the measurement and extraction arguments of Sections 5.2 and 5.3.

We call a three-message public-coin protocol two-strong-special sound with respect to a witness relation if any two accepting transcripts for the same public input and first message, with distinct challenge-response pairs, yield a witness in deterministic polynomial time. Here the witness is a nonzero short kernel vector for H, with the relation specified in Theorem 5.1. This combines special soundness with the unique-response property of Unruh [42], as used in [14].

Suppose two different response byte strings pass verification with a supplied challenge, V_{ID} of Definition 4.5, which accepts only canonical encodings, for the same public key, message, and first message (ρ, U). Their challenges may be equal. By Lemma 4.6, they give distinct reconstructed responses $\xi \neq \xi'$. Verification binds both recovered commitments to U, so by (10) both satisfy $H\xi = H\xi' = C(U) - \delta \pmod q$, and

$$H(\xi - \xi') = 0 \pmod q. \tag{29}$$

The verifier's secant check also bounds a weighted sum of squared coefficients. Combining it with the residual norm cap gives $\|\xi\|_2^2 \leq 11{,}350{,}941$ for every accepted response (Lemma B.1). The triangle inequality therefore gives

$$\|\xi - \xi'\|_2 < 6739 < q. \tag{30}$$

The integer vector $d = \xi - \xi'$ is nonzero. If it were zero modulo q, some nonzero coordinate would have absolute value at least q, contradicting $\|d\|_2 < q$. Thus it is also nonzero modulo q.

Theorem 5.1 (Two-strong-special soundness)**.** Fix a Pretzel-(4,4) public key pk, message digest μ, and first message (ρ, U), and let $H = [-b \mid A_0 \mid I_4]$ be determined by pk. Suppose $(\chi, Z) \neq (\chi', Z')$ and

$$V_{\mathrm{ID}}(\mathsf{pk}, \mu, \rho, U, \chi; Z) = V_{\mathrm{ID}}(\mathsf{pk}, \mu, \rho, U, \chi'; Z') = 1.$$

Then deterministic polynomial-time decoding and subtraction yield $d \in \mathcal{R}^9$ such that

$$Hd = 0 \pmod q, \qquad d \not\equiv 0 \pmod q, \qquad \|d\|_2 < 6739.$$

The statement includes $\chi = \chi'$ with $Z \neq Z'$.

Proof. If $\chi \neq \chi'$, the parity check forces $Z \neq Z'$; if $\chi = \chi'$, distinctness of the pairs already gives $Z \neq Z'$. Lemma 4.6 therefore gives distinct reconstructed responses ξ, ξ'. The extractor decodes them and returns $d = \xi - \xi'$. Equation (29) places d in the kernel of H, and (30) bounds its norm. Since $d \neq 0$ over the integers and $\|d\|_2 < q$, it is also nonzero modulo q. □

Hence d satisfies the short-kernel relation for $H = [-b \mid A_0 \mid I_m]$ with bound $\beta = 6739$ (Section 3). Normal-form Module-SIS needs a uniform matrix. The reduction of Theorem 5.6 obtains one by first removing the key-norm restriction, at a cost included in Δ_F of (F.1), and then replacing b by a uniform vector, at the cost of a Module-LWE advantage.

This subtraction argument is Gärtner's, who uses it to prove strong unforgeability [22]. Since verification with a supplied challenge checks the leading parities against the supplied challenge, accepted responses to different challenges have different bytes, so the theorem also covers the QROM extraction of Appendix F.

Remark 5.2 (Strong unforgeability). Strong unforgeability prevents an attacker who receives a valid signature from creating a different valid signature for the same message. If the forged signature reuses the message, salt, and commitment (μ, ρ, U) of a signature the attacker received, subtracting the two responses gives a short kernel vector by Theorem 5.1. Theorem 5.6 handles this case separately from forgeries on a fresh hash input, which go through measure-and-reprogram and sequential extraction; its bound includes both.

5.2 Measurement Hiding for Honestly Generated Keys

This section proves that detecting a response measurement would let an algorithm distinguish Module-LWE samples from uniform randomness. We first prove this for short vectors satisfying a fixed public linear equation. We then apply it to decoded signature responses. The adversary may keep classical data and quantum side information, including information from hash queries made before key generation. The reduction applies the result after signing has been replaced by simulation without the secret key.

Measuring a response removes interference between its distinct possible values. Applying random phases can also remove interference, a process called *dephasing*. The phases obtained from a Module-LWE sample only partially dephase the response. We calculate their effect on each pair of response values, then construct an interference experiment that recovers the original distinguishing gap divided by κ. The verifier's secant check bounds this loss.

The use of an LWE sample to apply phases to short preimages appears in Poremba's proof of Gaussian collapsing [38, Section 5.1, Theorem 4]. There, structured phases nearly preserve a Gaussian preimage state, while uniform phases measure it. We instead account for partial dephasing through the interference experiment. This applies to arbitrary accepted response states, including states entangled with the adversary's other registers.

The public key is honestly generated, but the adversary may prepare any key-dependent response state. This restriction is necessary for Pretzel (Remark B.3); Appendix B.1 compares the relevant collapsing notions.

Measurement hiding for short preimage superpositions. The core of the proof concerns any superposition of short preimages of one public target under $H = [A_h \mid I_m]$ with a uniform block $A_h \in \mathcal{R}_q^{m\times k}$. Identify $\mathcal{R}_q^d$ with $\mathbb{Z}_q^{256d}$ by listing coefficients, write $[x, y]_q = \sum_i x_i y_i \bmod q$, and define the adjoint $L^\dagger$ of a linear map L by $[L^\dagger r, x]_q = [r, Lx]_q$. It is the transpose with every $a(X)$ replaced by $a(X^{-1})$ and sends uniform matrices to uniform matrices. Fix ranks $k, m \geq 1$, noise parameters $\eta_{\mathrm{h}}, \eta_{\mathrm{r}} \geq 1$, and a loss $\kappa \geq 1$. For an integer vector $\xi \in \mathbb{Z}^{256(k+m)}$ with $\|\xi\|_\infty < q/4$ call the first $256k$ coordinates the head and the rest the residual, let $t_i = \eta_{\mathrm{h}}$ on the head and $t_i = \eta_{\mathrm{r}}$ on the residual, and define the secant sum

$$\Lambda(\xi) = \sum_i 2t_i \log_2 \sec \frac{2\pi\xi_i}{q}. \tag{31}$$

Definition 5.3 (Short-preimage measurement game)**.** Fix a public predicate $P(A_h, \tau, \xi)$ with a specified uniform reversible implementation, such that acceptance implies $\|\xi\|_\infty < q/4$, $H\xi = y \pmod q$, and $\Lambda(\xi) \leq \log_2 \kappa$. The predicate may enforce additional public restrictions. For a strict polynomial-time quantum adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ and a bit b, the game $\mathsf{PreColl}_b^{\mathcal{B}}$ samples $A_h \in \mathcal{R}_q^{m\times k}$ uniformly and gives it to $\mathcal{B}_1$, which outputs classical data τ containing a target $y \in \mathcal{R}_q^m$, a response register Z, and side information S. The game evaluates P reversibly, copies its acceptance bit, erases all temporary data, and measures only that bit, outputting zero on rejection. If $b = 1$ it measures Z. It gives (A_h, τ, Z, S) to $\mathcal{B}_2$ and outputs its bit. $\mathrm{Adv}_{\mathrm{PreColl}}(\mathcal{B})$ is the absolute difference of the two output probabilities. In Pretzel the integer secant table supplies the bound on Λ; the game need not decide an exact real-valued inequality.

Let $\mathrm{Adv}_{\mathsf{MLWE},\mathrm{aux}}$ be the advantage against decisional Module-LWE for the instance $(A_h^\dagger, A_h^\dagger s_{\mathrm{aux}} + e_{\mathrm{aux}})$ with $s_{\mathrm{aux}} \leftarrow \mathrm{CBD}_{\eta_r}^m$ and $e_{\mathrm{aux}} \leftarrow \mathrm{CBD}_{\eta_h}^k$ (k rows, m secret polynomials), and let $\varepsilon_{\mathrm{iso}}$ bound the change in either output probability when the implemented feature circuit is replaced by its ideal reference, including its completion registers. For the finite construction of Appendix B.4, this reference uses exact attenuations, untruncated feature directions, and the same response-dependent completion vectors in mutually orthogonal spaces.

Let ε_ϕ bound the change in either output probability caused by implementing the phase circuit, and let δ_u bound the total variation distance of the implemented mask distribution from uniform. Include their finite-bit sampling, reversible modular arithmetic, and phase-synthesis costs in the reduction's cost, as in Section F.5. Exact phase and mask operations correspond to $\varepsilon_\phi = \delta_u = 0$.

Theorem 5.4 (Measurement hiding for short preimage superpositions)**.** Suppose the public numerical parameters admit polynomial-time directed approximation to the requested precision, and the acceptance and approximate feature circuits have uniform polynomial cost, with output-probability error at most $\varepsilon_{\mathrm{iso}}$ as specified above. Every adversary $\mathcal{B}$ in Definition 5.3 then gives an auxiliary Module-LWE distinguisher with a constant number of adversary-circuit uses, plus the acceptance, phase, and feature-preparation costs, and

$$\mathrm{Adv}_{\mathrm{PreColl}}(\mathcal{B}) \le \kappa\ \mathrm{Adv}_{\mathsf{MLWE},\mathrm{aux}} + 2\kappa(\varepsilon_{\mathrm{iso}} + \varepsilon_\phi + \delta_u). \tag{32}$$

Dephasing along the equation. For fixed classical data τ, every accepted ξ solves $H\xi = y$. A vector $v \in \mathcal{R}_q^{k+m}$ applies the phase $\exp(2\pi i[v,\xi]_q/q)$ to $|\xi\rangle$, and $v = H^\dagger r$ applies the common, unobservable phase $\exp(2\pi i[r,y]_q/q)$ to every solution. Put $A = A_h^\dagger$, sample the auxiliary instance $w = As_{\mathrm{aux}} + e_{\mathrm{aux}}$ with $s_{\mathrm{aux}} \leftarrow \mathrm{CBD}_{\eta_r}^m$ and $e_{\mathrm{aux}} \leftarrow \mathrm{CBD}_{\eta_h}^k$, sample uniform $u \in \mathcal{R}_q^m$, and apply the phase indexed by

$$v = (Au - w, u) = H^\dagger(u - s_{\mathrm{aux}}) + (-e_{\mathrm{aux}}, s_{\mathrm{aux}}). \tag{34}$$

The first term is a common phase on the solution set. Averaging over the short vector $(-e_{\mathrm{aux}}, s_{\mathrm{aux}})$ multiplies each matrix entry of the state by a kernel,

$$\mathsf{T}(|\xi\rangle\langle\xi'|) = \Phi(\xi - \xi')\,|\xi\rangle\langle\xi'|, \qquad \Phi(d) = \prod_i \cos^{2t_i}\left(\frac{\pi d_i}{q}\right), \tag{35, 36}$$

because the characteristic function of CBD_η at d is $\cos^{2\eta}(\pi d/q)$. If instead w is uniform, then v is uniform and averaging removes every off-diagonal entry, since $|d_i| < q/2$ makes $d \equiv 0 \pmod q$ equivalent to $d = 0$. This is exactly measurement in the computational basis Δ.

Undoing the partial dephasing. For one coordinate with entries x, x' put $a = \pi x/q$ and $b = \pi x'/q$. Since $|x|, |x'| < q/4$, $|\tan a \tan b| < 1$ and $\cos^{-2t}(a-b) = \sec^{2t} a\ \sec^{2t} b\ (1 + \tan a \tan b)^{-2t}$ (37). Expanding the last factor in its binomial series gives feature vectors $\alpha_x, \beta_{x'}$ in ℓ^2 with

$$\langle \alpha_x, \beta_{x'} \rangle = \cos^{-2t}(a-b), \qquad \|\alpha_x\|^2 = \|\beta_x\|^2 = \sec^{2t}(2a). \tag{38, 39}$$

Tensor products over all coordinates, with $t = t_i$, give

$$K(\xi, \xi') = \Phi(\xi - \xi')^{-1} = \langle \alpha_\xi, \beta_{\xi'} \rangle. \tag{40}$$

By (38, 39) and (31), $\|\alpha_\xi\|^2 = \|\beta_\xi\|^2 = 2^{\Lambda(\xi)} \le \kappa$ for every accepted ξ.

Recovering the inverse by interference. The entrywise map $K \odot (\cdot)$ need not be a quantum operation, so we recover it as an amplitude. Choose two completion states $|\perp_A\rangle, |\perp_B\rangle$, orthogonal to each other and to the feature space, and define the unit vectors

$$|A_\xi\rangle = \frac{\alpha_\xi}{\sqrt{\kappa}} + \sqrt{1 - \frac{\|\alpha_\xi\|^2}{\kappa}}\,|\perp_A\rangle, \qquad |B_\xi\rangle = \frac{\beta_\xi}{\sqrt{\kappa}} + \sqrt{1 - \frac{\|\beta_\xi\|^2}{\kappa}}\,|\perp_B\rangle, \tag{41, 42}$$

so that $\langle B_{\xi'}|A_\xi\rangle = K(\xi, \xi')/\kappa$. The response-dependent completion vectors in the ideal reference used to define ε_{iso} satisfy the same identity.

Lemma 5.5 (Interference identity)**.** Let $\sigma = \sum_{\xi,\xi'} |\xi\rangle\langle\xi'| \otimes \sigma_{\xi,\xi'}$ be the subnormalized accepted state, possibly with side information; its trace is the acceptance probability. Prepare a control qubit in $|+\rangle$, and prepare $|A_\xi\rangle$ on branch zero and $|B_\xi\rangle$ on branch one, controlled on ξ. For any Hermitian M with $\|M\|_\infty \le 1$ on the other registers, the signed readout with effects $(I \pm X \otimes M)/2$ has the following expectation, weighted by the acceptance probability:

$$\mathbb{E}[X \otimes M] = \frac{1}{\kappa} \operatorname{Re} \operatorname{Tr}\big[M(K \odot \sigma)\big]. \tag{44}$$

Apply these identities blockwise with the classical registers (A_h, τ) retained in the state. Let E be the effect for output one of $\mathcal{B}_2$, controlled on those public registers, and let $g = \operatorname{Tr} E(\sigma - \Delta\sigma)$ denote the signed gap averaged over the joint distribution of the matrix and classical data τ. Then $\mathrm{Adv}_{\mathrm{PreColl}}(\mathcal{B}) = |g|$. The same readout is used in every block, so the absolute value is taken after this averaging. The distinguisher receives (A, w), gives $A_h = A^\dagger$ to $\mathcal{B}_1$, runs the acceptance test, applies the phase (34) with a fresh uniform u, and runs Lemma 5.5 with $M = 2E - I$. The accepted input state depends on A but not on w, which is not given to $\mathcal{B}_1$.

A real sample turns σ into $\mathsf{T}\sigma$ and a uniform sample into $\Delta\sigma$. Since $K \odot \mathsf{T}\sigma = \sigma$ by (40) and $K \odot \Delta\sigma = \Delta\sigma$ because $K(\xi, \xi) = 1$, identity (44) gives the difference of expectations $\kappa^{-1} \operatorname{Tr} M(K \odot (\mathsf{T}\sigma - \Delta\sigma)) = 2g/\kappa$. Implement the readout by measuring the control in the X basis to obtain $x \in \{-1, +1\}$, running $\mathcal{B}_2$ to obtain $b \in \{0, 1\}$, and outputting one exactly when $x(2b - 1) = +1$; on rejection output zero. This implements the effect $(I + X \otimes (2E - I))/2$ on the accepted branch. The two experiments have the same acceptance probability, so their output-probability difference is half the signed-expectation difference, and the distinguisher's advantage is $|g|/\kappa$.

Replacing the ideal feature circuit, phase circuit, and uniform mask by their finite implementations changes either output probability by at most $\varepsilon_{\text{iso}} + \varepsilon_\phi + \delta_u$. The triangle inequality therefore gives $|g| \le \kappa\,\mathrm{Adv}_{\mathsf{MLWE},\mathrm{aux}} + 2\kappa(\varepsilon_{\text{iso}} + \varepsilon_\phi + \delta_u)$. This proves (32). Appendix B.3 supplies the ideal identities, and Section F.5 specifies the finite phase and mask implementations for Pretzel-(4,4).

Application to the signature response. For fixed classical data, Lemma 4.6 gives a bijection between accepted canonical signature bytes and their reconstructed responses $\xi = (z_h, r_e)$, with $H\xi = C(U) - \delta \pmod q$. The decoder uses the supplied challenge of Definition 4.5, without another hash query. By Remark B.2, this bijection can be run reversibly without leaving a copy of the signature. Thus measuring accepted bytes is equivalent to measuring the response, with the same rejection branch and with reference registers retained.

The verifier's coefficient and secant checks give the theorem's numerical acceptance conditions. For Pretzel-(4,4), $\eta_{\mathrm{h}} = 6$ and $\eta_{\mathrm{r}} = 2$, which give the weights 12 and 4 in the secant sum; the auxiliary instance therefore has CBD_2^4 secret and CBD_6^5 error. The classical data τ comprise the key, message, salt, commitment, and challenge; only the response register differs between the two experiments. Appendix B.5 applies the theorem to honestly generated keys by replacing the public block with a uniform matrix and charging the key Module-LWE advantages and setup-selection loss.

Appendix C.2 gives a sharper one-sided bound on the extraction probability, using an operation that scales the amplitudes of the same reconstructed responses.

5.3 From Forgery to Lattice Problems

The following theorem bounds the ordinary (EUF) and strong (SUF) forgery probabilities for Pretzel-(4,4) in the QROM. Ordinary unforgeability requires a forgery on an unsigned message; strong unforgeability also counts a new signature on a previously signed message.

Hypotheses. Keys are honestly generated and immutable, with a fixed honest application context. Key-generation and signing requests are classical, while the adversary may query the public hash in superposition. Signing requests in superposition are outside this model; they give key-recovery attacks on early versions of Dilithium, though not on ML-DSA [7]. Verification and preparation requests through the public interface are also classical. Hashing is modeled as an ideal extendable-output function. The experiment excludes secret-key export, corruption, private inverse access, entropy faults and physical leakage, and assumes functional integer, buffer and FFI semantics (Section F.1). We use the complete experiment and functional contracts of Appendix F, including Assumption F.1: the sampler and interfaces must have the specified complete output distributions and retained state, including retries, failed attempts, setup failures, and replay of repeated attempts. These conditions concern the full computation, not only successful signatures.

The cost of the constructed lattice algorithms includes all their work and storage, including preprocessing, forward and inverse operations, and finite simulation of the random oracles; they use no external random oracle. The matrix interface and its cost exclusions are those of Assumption F.1. The algorithms inherit the adversary's advice class, with advice independent of the fresh oracle tables and experiment coins. Hardness for efficiently preparable advice charges its preparation; arbitrary independent quantum advice requires hardness in that class, as specified in Section F.1.

Notation. Write Δ_F for the full loss in comparing real and simulated signing in (F.1), including sampling, programming, and removal of the key-norm restriction. Write Δ_μ for the probability of a message-prehash collision in the unconditioned ideal simulation (Section F.4). Let $q_B = 2h + r + V$, where h, r, and V bound direct typed-challenge queries, raw-prefix queries, and public verification or preparation requests, respectively, as counted in Section F.2. Let $1 \le K \le 2^{64}$ bound all potential key-generation calls, including failed and unreached positions.

For a fixed trial cap $J \ge 1$, finite uniform selection among n choices succeeds with probability $\lambda_n = 1 - (1 - n/2^{\lceil \log_2 n \rceil})^J$, with $\lambda_1 = 1$. Put $K_* = K/\lambda_K$ and $\lambda_{\mathrm{MR}} = \lambda_{2q_B+1}$; Section F.4 specifies the selection procedure. For $[x]_+ = \max\{x, 0\}$ and $C = 2^{255}$ folded challenges, put

$$f_C(x) = \frac{x(x - 1/C)_+^2}{1 - 1/C}, \qquad 0 \le x \le 1.$$

The secant check gives $\kappa = 2^{964091/32768}$, and the extracted Module-SIS vectors have norm less than 6739 (Lemma B.1). The error δ_{sim} comes from finite oracle simulation (Section F.2); $\varepsilon_{\mathrm{iso}}$, ε_ϕ, and δ_u are the errors in the feature circuit, phase rotations, and auxiliary coin sampling, respectively (Section F.5).

Theorem 5.6 (Finite QROM bound for Pretzel)**.** Under the hypotheses above, let ϵ be the EUF or SUF forgery probability. Let s, with subscripts, denote the success probabilities of the constructed ordinary normal-form SIS algorithms, and a, with subscripts, the advantages of the constructed ordinary key or auxiliary MLWE algorithms, in the advice class inherited from the adversary. Their costs include all work and storage. Put $P_{\mathrm{pair}} = 0$ for EUF, and for SUF put

$$P_{\mathrm{pair}} = K_*(s_{\mathrm{pair}} + a_{\mathrm{key,pair}} + 3\delta_{\mathrm{sim}}).$$

Put $\epsilon_0 = [\epsilon - \Delta_F - \Delta_\mu - P_{\mathrm{pair}}]_+$. Then

$$f_{2^{255}}\left(\frac{\lambda_{\mathrm{MR}}\epsilon_0}{(2q_B+1)^2}\right) \le K_*\big[s_{\mathrm{ext}} + a_{\mathrm{key,rel}} + a_{\mathrm{key},0} + a_{\mathrm{key},1} + \kappa a_{\mathrm{aux}} + (7+2\kappa)\delta_{\mathrm{sim}} + 2\kappa(\varepsilon_{\mathrm{iso}} + \varepsilon_\phi + \delta_u)\big]. \tag{45}$$

The proof is in Appendix F, Section F.4.

The bound relates forgery probability to Module-SIS success and key and auxiliary Module-LWE advantages, after subtracting the signing-simulation loss Δ_F, the collision loss Δ_μ, and the SUF pair term. The factor κ from the secant check weights the auxiliary Module-LWE advantage and the associated simulation and precision errors; it does not multiply the key advantages or the Module-SIS term in this theorem. The query loss $(2q_B+1)^2$ and the finite-selection probability λ_{MR} come from measure-and-reprogram, while K_* accounts for selecting the attacked setup position.

The bound at the proposed parameters. Bound (45) gives no useful concrete bound at the proposed parameters. Suppose every success probability and advantage on its right-hand side is at most 2^{-220}, $K_* = \lambda_{\mathrm{MR}} = 1$, the simulation and precision errors vanish, and $\Delta_F = \Delta_\mu = P_{\mathrm{pair}} = 0$. Then the bound forces $\epsilon < 1/2$ only for $q_B < 2^{30.3}$. Parameters are therefore set from attack estimates (Section 6).

How the proof uses a forgery. The public simulator in Appendix E answers signing requests without the secret key by choosing a challenge, generating a response, recovering its commitment, and programming the challenge hash. The comparison includes the retained quantum state and aborts. After excluding message-prehash collisions, a strong forgery at a previously programmed challenge input supplies a second response directly; P_{pair} bounds this event. Measure-and-reprogram [13] turns the other forgeries into a prover with success at least $g = \lambda_{\mathrm{MR}}\epsilon_0/(2q_B+1)^2$. The extractor obtains one response, reverses the prover, and obtains a response to a new uniform folded challenge, retaining the unused raw challenge bit. Before measuring the first response, the probability of two accepting responses with different challenges is at least $f_C(g)$. Measurement hiding permits recording that response; subtracting the two responses gives the short vector of Theorem 5.1.

Refinements. Corollary F.5 bounds this extraction event using the response filter of Lemma C.2. Its coefficient $\sqrt{\kappa}$ weights both the auxiliary Module-LWE advantage and Module-SIS success, with separate tester costs and errors. For ordinary unforgeability, Corollary F.6 instead uses a return test with quadratic dependence on prover success, under the additional coherent challenge interface specified there: the post-challenge computation supports superposed challenge values and its clean inverse. Appendix C.2 proves these refinements and gives their complete costs.

6 Concrete Security Analysis

We assess concrete security by estimating the work required by known lattice attacks.

For Pretzel-(4,4), key-recovery estimates use the Module-LWE instance with CBD_1 keys at $(q, \ell, m) = (7681, 4, 4)$ and ignore the key-norm restriction (3); Appendix G shows that the restriction can increase a fixed key-recovery algorithm's success probability only by a negligible relative amount. We have not verified that the compiled implementation runs in constant time or erases secret data.

6.1 Comparison with ML-DSA

ML-DSA's quantum analysis assumes, in addition to Module-LWE and Module-SIS, that SelfTargetMSIS is hard. This short-vector problem requires the solution to match a challenge obtained by hashing the message and the reconstructed commitment. Classically, Module-SIS reduces to it [26]: a SelfTargetMSIS solver yields a Module-SIS solver. Jackson, Miller, and Wang derive its quantum hardness from Module-LWE, but only for parameters that exclude standardized ML-DSA [25, 35]. Pretzel's proof avoids this assumption: it extracts two responses and subtracts them, using measurement hiding under Module-LWE to do so against a quantum forger.

One cost of Pretzel's design is that its proof must bound finite-precision errors in sampling and their accumulation over signing retries. ML-DSA's uniform integer masks and exact rejection tests avoid this [35].

6.2 Attack estimates

Matched lattice-attack estimates. Table 5 pairs Pretzel-(3,3), -(4,4), and -(5,5) with ML-DSA-44, -65, and -87, respectively. The estimates use the full Lattice Estimator LWE/SIS portfolio at one revision, under the classical cost models MATZOV [32] and Alkim, Ducas, Pöppelmann, and Schwabe [2] (abbreviated ADPS16). The cost of the dual-hybrid attack is minimized over all of its parameters, including the number of samples, without the estimator's early stopping rules.

In both models, Pretzel-(4,4)'s key-recovery cost exceeds ML-DSA-65's by 14 to 17 bits and falls 48 to 49 bits below ML-DSA-87's. Its Module-SIS estimate exceeds that for ML-DSA-87's strong-unforgeability instance by 20 to 24 bits.

Pretzel-(3,3)'s key-recovery cost exceeds ML-DSA-44's by 16 to 19 bits and falls below ML-DSA-65's, consistent with its category-2 target.

The auxiliary Module-LWE instance appears only in the security proof (Section 5.2). The scheme never samples it, but the proof assumes it is hard.

Public-key size and the auxiliary instance. At category 3, Pretzel's public key holds four polynomials with 13-bit coefficients, while SHUTTLE-180's holds three with 14-bit coefficients. Polynomial storage explains most of the public-key size difference in Table 3; the formats also have different overheads.

Pretzel uses four public polynomials to increase the estimated hardness of the auxiliary Module-LWE instance. Its secret dimension is larger than Pretzel-(3,3)'s. The two auxiliary instances use the same noise distributions. Table 5 places Pretzel-(3,3)'s auxiliary attack costs below ML-DSA-65's key-recovery costs under both models.

For Pretzel-(5,5), estimated key-recovery costs exceed ML-DSA-87's by 6.7 to 10.0 bits across the four cost models of Table 10.

Generic forgery search. Forgery security is also limited by the size of the challenge space. Fix one valid signature and its context, then search for a different message whose recomputed challenge equals the signature's leading parities. The response, salt, and recovered commitment stay fixed, so all other verification checks continue to hold. For Pretzel-(4,4) and Pretzel-(5,5), which use the uniform 2^{255}-element challenge family, an independent trial succeeds with probability 2^{-255}. Classical search therefore takes order 2^{255}

Table 5: Selected attack costs, $\log_2$ of modeled operations, computed with the Lattice Estimator under the full attack portfolio. ML-DSA key rows use the unrounded public relation. ML-DSA SIS rows are the Module-SIS instance used for strong unforgeability (infinity norm). Pretzel SIS rows use normal-form Module-SIS with squared Euclidean radius 47,801,092 for $(3, 3)$ and 58,848,400 for $(5, 5)$ (radii below 6913.835 and 7671.272), and radius 6739 for $(4, 4)$ (Theorem 5.1). The auxiliary MLWE rows concern the measurement-hiding step. Rows are grouped by role; ADPS16 is the classical model. ML-DSA's self-target relaxation is estimated for ML-DSA-87 only, in Table 10. The costs exclude reduction losses.

Role	Model	Instance	$(3, 3)$	ML-DSA-44	$(4, 4)$	ML-DSA-65	$(5, 5)$	ML-DSA-87
Key recovery	MATZOV	key MLWE	164.5	148.0	218.1	203.8	273.5	266.9
	ADPS16	key MLWE	141.6	122.6	197.3	179.9	254.1	244.9
Extra assumption in the quantum proof	MATZOV	auxiliary MLWE	189.0		251.7		315.8	
	ADPS16	auxiliary MLWE	166.8		231.6		297.6	
Forgery	MATZOV	Module-SIS	209.6	149.3	296.6	201.5	374.3	276.1
	ADPS16	Module-SIS	186.9	121.8	277.4	176.1	358.0	253.5

trials, while ideal quantum search takes order $2^{127.5}$ oracle evaluations [24]. Pretzel-(3,3) uses a smaller challenge family; its corresponding search cost is given in Appendix H.

For ML-DSA, fix a valid response and hint and search for a different message whose new commitment hash expands to the same challenge polynomial. Replace the commitment hash in the signature by this new value. The recovered commitment stays fixed, and the final hash equality then holds by construction [35, Algorithms 8 and 29]. The signed challenge polynomials number $2^\tau \binom{256}{\tau}$. For ML-DSA-65, $\tau = 49$, so ideal uniform-challenge search has quantum query exponent approximately 112.7, regardless of the 48-byte length of the stored hash. These are ideal query counts, not gate counts. Forgery security is limited by this search cost as well as by the lattice-attack costs above.

Appendix G gives the bound for the key-norm restriction, the estimator inputs, and Module-SIS estimates at the sharper extraction radius 6739.

7 Implementation and Performance

We implement Pretzel-(3,3), Pretzel-(4,4), and Pretzel-(5,5) and compare them with AWS-LC's corresponding ML-DSA implementations on Apple silicon under macOS. The reference implementation, with its certificates and the benchmark that compares it with ML-DSA, is available at `https://github.com/a16z/pretzel`. We report Pretzel-(4,4) in detail and summarize the other two parameter sets. Signing times include fresh randomness and every retry. Key generation runs once per key. Every parameter set tests candidate keys and chooses the first dither whose bounded expansion succeeds. A prepared public key has already been unpacked from its stored form into the matrix and offset used in verification.

7.1 Signature sizes and performance

The 1361-byte Pretzel encoding reduces signature size by 59% relative to ML-DSA-65. On macOS (Apple M2 Pro), signing is about 7% faster in mean time and verification including public-key preparation about 44% faster. With a prepared public key, verification is also faster than ML-DSA-65's on these short messages, by a smaller margin. Tables 6 and 7 give the sizes and timings. Each printed ratio is the geometric mean over runner jobs of matched ratios, and the printed times are indicative means, so a ratio need not equal the quotient of the two times beside it.

Table 6: Serialized sizes in bytes for Pretzel-(4,4) and ML-DSA-65. Secret keys use the serialized formats measured here, rather than a key-generation seed.

Object	Pretzel-(4,4)	ML-DSA-65
Signature	1361	3309
Public key	1742	1952
Secret key	654	4032

Table 7: Pretzel-(4,4) and ML-DSA-65 mean thread CPU time in microseconds. Time ratios compare Pretzel-(4,4) with ML-DSA-65, so values below one favor Pretzel. Brackets give 95% intervals. Verify prepared checks a signature with a public key that has already been prepared, that is, unpacked into the matrix and offset used in verification; Prepare public key is that unpacking; Verify with preparation does both, starting from the stored key.

Operation	Pretzel-(4,4)	ML-DSA-65	Time ratio (interval)
Sign (prepared secret key)	56.697	60.894	0.9310 [0.9247, 0.9375]
Verify with preparation	16.914	30.480	0.5551 [0.5528, 0.5574]
Prepare public key	12.331	25.187	0.4884 [0.4834, 0.4955]
Verify prepared	4.834	5.347	0.9046 [0.8879, 0.9182]
Generate key (KG)	44.51	33.23	1.3394 [1.3330, 1.3458]

Messages are 32 bytes and contexts are empty. Key generation and signing draw fresh entropy from the operating system. Verification with the public key already unpacked checks one signature produced before timing. Verification including public-key preparation starts from the serialized public key and includes releasing the temporary prepared data. Preparation measurements also include cleanup. Signing uses a secret key already in its prepared form.

Pretzel-(3,3) measurements against AWS-LC's ML-DSA-44 appear in Appendix H: 1160-byte signatures, against 2420 bytes, with faster signing and verification. Pretzel-(5,5) measurements against ML-DSA-87 appear in Section I.4. For Pretzel-(5,5), key generation on macOS takes about 0.055 ms per key, about the same as the 0.055 ms of AWS-LC's ML-DSA-87. For Pretzel-(4,4), the corresponding ratio against ML-DSA-65 is in Table 7.

The cost of key generation is amortized over the signatures made with each key and may matter on constrained devices or in applications that generate many keys.

7.2 Implementation techniques

The measured Pretzel-(4,4) implementation, with signatures of 1361 bytes, targets AArch64 with NEON and SHA3 instructions and uses the following techniques.

Reusing inner products. Each transport step needs the weighted inner product $\langle z, X^j s\rangle_W$ of the current response with a rotation of the secret, and recomputing it would take a pass over z per step. Instead, the signer computes all 256 initial values $\langle y, X^j s\rangle_W$ together using fast polynomial multiplication. It reuses precomputed transforms of the secret key and uses a modulus large enough to recover the integer results exactly. For each key it also precomputes $G_k = \langle s, X^k s\rangle_W$. The same Gram matrix of the secret's rotations appears in BLISS [15, 16]. Because every update is a rotation of the same secret, a later inner product is its initial value plus a signed sum of Gram entries from earlier updates, with negacyclic wraparound signs. After the transport loop and its deferred checks, the signer forms $z = y + ds$, where d is the polynomial of signed updates, then applies the public response checks of Table 4.

Step decisions and mask sampling. The signer approximates the transport probabilities of Section 4.3 using finite-precision integer comparisons, with higher-precision checks deferred until later in the attempt. Most mask coefficients are determined from a few random bits, with more bits used for choices not yet determined. Signing reuses hash computations and the expanded dither prepared for the key. Appendix D specifies the finite sampler and the limits that can cause an attempt to fail.

Encoding. Signing checks that the response fits in the encoding with scans that follow a fixed sequence and runs Rice packing only after the response has passed all checks. Packing can take variable time.

Public-key preparation and verification. The public key stores b in the NTT domain, and A_0 is sampled directly as NTT coordinates, so preparing a public key requires no NTTs. ML-DSA-65 instead expands 30 matrix polynomials, against Pretzel's 16, and transforms its public vector. Verification then processes each row of $x = A_h z_h + \delta$ in one fused NEON kernel that computes the matrix-vector product, the inverse NTT, the label recovery, the residual caps, and the Euclidean and secant sums, the last two through a single table lookup per coefficient.

Side channels. The signer uses lookups in processor registers to limit memory accesses that depend on secret data. It also enables a processor mode intended to limit timing differences caused by data. Failed steps set a flag. The attempt completes all 128 steps and their checks before deciding whether to restart. The implementation follows constant-time coding practice for secret-dependent operations. We have not evaluated power or electromagnetic leakage. Pretzel makes no masking claim: its signer uses rejection sampling, and whether a step restarts depends on the secret. Raccoon [10] instead removes rejection sampling, so that signing can be masked with d shares at overhead $O(d \log d)$, with a security proof in the probing model. It pays with signatures about ten times larger, 14,544 bytes at category 3 [9].

Acknowledgments

We thank Quang Dao, Joshua Limbrey and Hao Yan for a detailed review of the paper and the implementation. We used AI tools with editing, checking and experiments; all content was reviewed by the authors.

Disclosure

All three authors are affiliated with a16z frontier research. a16z is an investor in various blockchain-based platforms, as well as in the crypto ecosystem more broadly. General a16z disclosures are available at https://www.a16z.com/disclosures/.

References

[1] Martin R. Albrecht et al. Lattice estimator, 2026. Software, revision 53da5982, accessed September 2026.

[2] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange – a new hope. In *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association, 2016. Cryptology ePrint Archive, Paper 2015/1092.

[3] Manuel Barbosa, Gilles Barthe, Christian Doczkal, Jelle Don, Serge Fehr, Benjamin Grégoire, Yu-Hsuan Huang, Andreas Hülsing, Yi Lee, and Xiaodi Wu. Fixing and mechanizing the security proof of Fiat–Shamir with aborts and Dilithium. In *Advances in Cryptology – CRYPTO 2023*, volume 14085 of *LNCS*, pages 358–389. Springer, 2023.

[4] André Chailloux and Johanna Loyer. Lattice sieving via quantum random walks. In *Advances in Cryptology – ASIACRYPT 2021*, 2021. arXiv:2105.05608.

[5] Jung Hee Cheon, Hyeongmin Choe, Julien Devevey, Tim Güneysu, Dongyeon Hong, Markus Krausz, Georg Land, Marc Möller, Damien Stehlé, and MinJune Yi. HAETAE: Shorter lattice-based Fiat-Shamir signatures. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2024(3):25–75, 2024. Cryptology ePrint Archive, Paper 2023/624.

[6] Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. On the compressed-oracle technique, and post-quantum security of proofs of sequential work. https://ir.cwi.nl/pub/31044/31044.pdf, 2021. Full version of the EUROCRYPT 2021 article; Theorem 5.29.

[7] Carlos Cid, David Elkouss, and Manuel Goulão. Superposition key-recovery attacks on Dilithium and Fiat-Shamir signature schemes. Cryptology ePrint Archive, Paper 2026/2258, 2026.

[8] Marcel Dall'Agnol and Nicholas Spooner. On the necessity of collapsing for post-quantum and quantum commitments. In *18th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2023)*, volume 266 of *LIPIcs*, 2023.

[9] Rafael del Pino, Thomas Espitau, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, Mélissa Rossi, and Markku-Juhani Saarinen. Raccoon: A side-channel secure signature scheme, 2023. Submission to the NIST call for additional post-quantum signatures.

[10] Rafaël del Pino, Shuichi Katsumata, Thomas Prest, and Mélissa Rossi. Raccoon: A masking-friendly signature proven in the probing model. In *Advances in Cryptology – CRYPTO 2024*, Lecture Notes in Computer Science, pages 409–444. Springer, 2024. Cryptology ePrint Archive, Paper 2024/1291.

[11] Julien Devevey, Pouria Fallahpour, Alain Passelègue, and Damien Stehlé. A detailed analysis of Fiat–Shamir with aborts. In *Advances in Cryptology – CRYPTO 2023*, volume 14085 of *LNCS*, pages 327–357. Springer, 2023.

[12] Julien Devevey, Alain Passelègue, and Damien Stehlé. G+G: A Fiat-Shamir lattice signature based on convolved Gaussians. In *Advances in Cryptology – ASIACRYPT 2023, Part VII*, volume 14444 of *LNCS*, pages 37–64. Springer, 2023. Cryptology ePrint Archive, Paper 2023/1477, revised November 13, 2023.

[13] Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round Fiat–Shamir and more. In *Advances in Cryptology – CRYPTO 2020*, 2020. arXiv:2003.05207.

[14] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat–Shamir transformation in the quantum random-oracle model. In *Advances in Cryptology – CRYPTO 2019*, 2019. Cryptology ePrint Archive, Paper 2019/190.

[15] Léo Ducas. Accelerating Bliss: the geometry of ternary polynomials. Cryptology ePrint Archive, Paper 2014/874, 2014.

[16] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal Gaussians. In *Advances in Cryptology – CRYPTO 2013*, volume 8042 of *LNCS*, pages 40–56. Springer, 2013.

[17] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(1):238–268, 2018.

[18] Jarek Duda. Asymmetric numeral systems: Entropy coding combining speed of Huffman coding with compression rate of arithmetic coding. arXiv:1311.2540, 2013.

[19] Pouria Fallahpour, Serge Fehr, and Yu-Hsuan Huang. Tighter quantum security for Fiat–Shamir-with-aborts and hash-and-sign-with-retry signatures, 2025. Cryptology ePrint Archive, Paper 2025/985, revised April 2026.

[20] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-fourier lattice-based compact signatures over NTRU. Specification v1.2, NIST Post-Quantum Cryptography Standardization, Round 3, 2020.

[21] Joel Gärtner. NTWE: A natural combination of NTRU and LWE. In *Post-Quantum Cryptography – PQCrypto 2023*, LNCS. Springer, 2023. Cryptology ePrint Archive, Paper 2023/663.

[22] Joel Gärtner. Compact lattice signatures via iterative rejection sampling. In *Advances in Cryptology – CRYPTO 2025*, 2025. Cryptology ePrint Archive, Paper 2024/2052, revised May 2025.

[23] Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the QROM. In *Advances in Cryptology – ASIACRYPT 2021*, LNCS, pages 637–667. Springer, 2021.

[24] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pages 212–219, 1996.

[25] Kelsey A. Jackson, Carl A. Miller, and Daochen Wang. Evaluating the security of CRYSTALS-Dilithium in the quantum random oracle model. In *Advances in Cryptology – EUROCRYPT 2024*, volume 14656 of *LNCS*, pages 418–446. Springer, 2024. Cryptology ePrint Archive, Paper 2023/1968.

[26] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In *Advances in Cryptology – EUROCRYPT 2018*, volume 10822 of *LNCS*, pages 552–586. Springer, 2018.

[27] Jiahui Liu, Hart Montgomery, and Mark Zhandry. Another round of breaking and making quantum money: How to not build it from lattices, and more. In *Advances in Cryptology – EUROCRYPT 2023*, LNCS, pages 611–638. Springer, 2023.

[28] Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. In *Advances in Cryptology – CRYPTO 2019*, LNCS. Springer, 2019. Cryptology ePrint Archive, Paper 2019/262.

[29] Alex Lombardi, Fermi Ma, and Nicholas Spooner. Post-quantum zero knowledge, revisited: Or how to do quantum rewinding undetectably. In *63rd IEEE Symposium on Foundations of Computer Science (FOCS 2022)*, 2022. arXiv:2111.12257.

[30] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, 2009.

[31] Vadim Lyubashevsky. Lattice signatures without trapdoors. In *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, 2012.

[32] MATZOV. Report on the security of LWE: Improved dual lattice attack. Zenodo, 2022.

[33] Yifan Ming, Jipeng Zhang, Zihan Liu, Guofeng Tang, Pengfei Chen, Yutao Sun, Si Gao, Cong Zhang, and Long Chen. Towards practical iterative rejection sampling: A compact and efficient signature over module lattices. Cryptology ePrint Archive, Paper 2026/1991, 2026. Accepted at ASIACRYPT 2026.

[34] National Institute of Standards and Technology. Digital signature standard (DSS). FIPS PUB 186-5, 2023.

[35] National Institute of Standards and Technology. Module-lattice-based digital signature standard. FIPS PUB 204, 2024.

[36] National Institute of Standards and Technology. Stateless hash-based digital signature standard. FIPS PUB 205, 2024.

[37] Ray Perlner. New plan for FN-DSA. NIST PQC forum (pqc-forum) post, for the FIPS 206 team, 2026. Posted 28 September 2026.

[38] Alexander Poremba. Quantum proofs of deletion for learning with errors. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *LIPIcs*, pages 90:1–90:14, 2023. Full version: arXiv:2203.01610, Section 5.1 and Theorem 4.

[39] Thomas Pornin. A RAM-efficient implementation of Falcon, 2026. Cryptology ePrint Archive, Paper 2026/1915.

[40] Robert F. Rice and James R. Plaunt. Adaptive variable-length coding for efficient compression of spacecraft television data. *IEEE Transactions on Communication Technology*, 19(6):889–897, 1971.

[41] SHUTTLE project. SHUTTLE-NIST: Reference implementation, 2026. Algorithm parameters and reported implementation sizes. Accessed 25 September 2026.

[42] Dominique Unruh. Quantum proofs of knowledge. In *Advances in Cryptology – EUROCRYPT 2012*, 2012. Cryptology ePrint Archive, Paper 2010/212.

[43] Dominique Unruh. Collapse-binding quantum commitments without random oracles. In *Advances in Cryptology – ASIACRYPT 2016*, volume 10032 of *LNCS*. Springer, 2016. Full version: ePrint 2016/508.

[44] Sebastiano Vigna. Quasi-succinct indices. In *Proceedings of WSDM*, pages 83–92, 2013.

[45] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *Advances in Cryptology – CRYPTO 2019*, 2019. Full version: Cryptology ePrint Archive, Paper 2018/276.

[46] Jipeng Zhang, Pengfei Chen, Long Chen, Cong Zhang, and Jiaheng Zhang. Lithium: Making iterative rejection sampling practical for compact lattice signatures, 2026. Cryptology ePrint Archive, Paper 2026/1790, 24 August 2026.

A Proofs for the Transport Sampler

of Lemma 4.1. For $u \geq 0$ the series (15) alternates with terms decreasing in magnitude, so $T(u, b) \in [0, 1]$. Each expression for f or g in (17)–(18) is a nonnegative multiple of such a value, which proves positivity. This is Gärtner's argument [22]. It depends only on the two numbers $a = \langle z, v\rangle_W$ and $b = E(v)$, so it applies to our weighted norm. For the sum, the recurrence (A.5) below gives $\widehat{f} + \widehat{g} = 1$ whenever $a \geq b$ or $a < -b$. For $-b \leq a < b$, writing out both series and reindexing gives

$$\widehat{f}(a, b) + \widehat{g}(a, b) = 1 - \sum_{j\in\mathbb{Z}} (-1)^j \theta^{j^2 b - 2ja}. \tag{A.1}$$

Put $t = -b \log\theta > 0$ and $u = a/b$, so that $-1 \leq u < 1$. Poisson summation gives

$$\sum_{j\in\mathbb{Z}} (-1)^j \theta^{j^2 b - 2ja} = e^{tu^2} \sqrt{\frac{\pi}{t}} \sum_{k\in\mathbb{Z}} e^{-\pi^2 (k+1/2)^2/t} \cos\bigl(\pi(2k+1)u\bigr), \tag{A.2}$$

and the Jacobi imaginary transformation gives

$$\mathcal{M}(b) - 1 = e^t \vartheta_4(0, e^{-t}) = e^t \sqrt{\frac{\pi}{t}} \sum_{k\in\mathbb{Z}} \exp\left(-\frac{\pi^2 (k+1/2)^2}{t}\right). \tag{A.3}$$

Since $u^2 \leq 1$, the absolute value of the sum in (A.2) is at most $e^t \sqrt{\pi/t} \sum_k e^{-\pi^2(k+1/2)^2/t} = \mathcal{M}(b) - 1$ by (A.3), with equality at $u = -1$, that is, at $a = -b$. This proves (19) with the exact supremum (20). Gärtner bounds the same quantity by a majorant M_α obtained from Poisson summation [22]; (20) is instead the exact supremum for our sampler.

For monotonicity, each term of (A.3) has logarithmic derivative in t

$$1 - \frac{1}{2t} + \frac{\pi^2 (k+1/2)^2}{t^2} > 0.$$

This is positive because $\pi^2(k+1/2)^2 \geq \pi^2/4 > 1/16 \geq t/2 - t^2$. Since t increases with b, $\mathcal{M}$ increases with b. Accepted keys have $b = E(s) \leq 230{,}053$, so the certified bound (A.6) gives $\mathcal{M}(b) < 65{,}612/65{,}536 = M$. Dividing (19) by M proves the claim. □

of Lemma 4.2. Gärtner's proof uses the identity

$$\|z + kv\|^2 = \|z\|^2 + 2k\langle z, v\rangle + k^2 \|v\|^2.$$

Our weighted squared norm satisfies the same identity

$$E(z + kv) = E(z) + 2k\langle z, v\rangle_W + k^2 E(v). \tag{A.4}$$

The series (15) satisfies the recurrence

$$T(a, b) = 1 - \theta^{2a+b} T(a + b, b), \tag{A.5}$$

which follows by separating the $k = 0$ term and reindexing $k \mapsto k + 1$. Let $a' = \langle x, v\rangle_W$. By (A.4), $p(x \pm v) = p(x)\, \theta^{b \pm 2a'}$, $\langle x + v, v\rangle_W = a' + b$, and $\langle x - v, v\rangle_W = a' - b$. Multiplying (22) by $M/p(x)$, we must show

$$\theta^{b+2a'} \widehat{f}(a' + b, b) + \theta^{b-2a'} \widehat{g}(a' - b, b) = 1.$$

If $a' \geq 0$, then $a' + b \geq b$ and $a' - b \geq -b$, so by (17)–(18) the left side equals $\theta^{2a'+b} T(a' + b, b) + T(a', b)$, which is 1 by (A.5) with $a = a'$. If $a' < 0$, then $a' + b < b$ and $a' - b < -b$, so the left side equals $T(-a', b) + \theta^{-2a'+b} T(b - a', b)$, which is 1 by (A.5) with $a = -a'$. The identity uses only $b > 0$, so it holds for every key. The key-norm restriction (3) is needed only to make f and g probabilities (Lemma 4.1). □

A.1 Proof of Theorem 4.3

Let $v = X^j s$ and recall the hypothesis $c'_j = 0$. For a parity vector β, put $Z_\beta = \sum_{\mathrm{parity}(z_0)=\beta} p(z)$, so that $D_\beta = p/Z_\beta$ on its parity class. Every x with $\mathrm{parity}(x_0) = c' \oplus e_j$ has exactly two preimages, $x + v$ and $x - v$, in the class c'. By (22), its output mass is

$$\frac{p(x)}{MZ_{c'}} = \frac{Z_{c'\oplus e_j}}{MZ_{c'}} D_{c'\oplus e_j}(x).$$

The coordinate weights factor, and coordinate j changes from even to odd. All other normalizing factors cancel, giving $Z_{c'\oplus e_j}/Z_{c'} = Z_{\mathrm{odd}}/Z_{\mathrm{even}}$. The displayed coefficient is therefore λ by (23), proving (24).

Now fix a challenge c independently of the initial D_0 sample. Each of its w selected positions is processed once, in the fixed public order, so its parity is zero before that step. Induction gives the unnormalized surviving measure $\lambda^w D_c$, as in (25).

The $128 - w$ independent dummy tests multiply this measure by λ^{128-w}. The resulting measure is $\lambda^{128} D_c$, which proves (26).

Commitment-dependent challenges. For a commitment value U let $\mathcal{Z}_U = \{z : Q(Hz + \delta) = U\}$, and for a measure ν write $\mathbf{1}_{\mathcal{Z}_U}\nu$ for its unnormalized restriction to $\mathcal{Z}_U$. Fix the public key, message digest μ, salt ρ and challenge oracle, so that signing chooses the challenge $c(U) \in C_{255}$ as a function of the commitment. Let K_c be the composition of the transport kernels of c, in the fixed public order, and its $128 - \mathrm{wt}(c)$ dummy tests. Since $Hv = 0$ for $v = X^j s$, $\mathbf{1}_{\mathcal{Z}_U}(x \pm v) = \mathbf{1}_{\mathcal{Z}_U}(x)$, so restriction to $\mathcal{Z}_U$ commutes with every K_v and with the dummy tests. By (26),

$$(\mathbf{1}_{\mathcal{Z}_U} D_0) K_c = \lambda^{128}\, \mathbf{1}_{\mathcal{Z}_U} D_c,$$

and the ideal surviving response measure is $\lambda^{128} \sum_U \mathbf{1}_{\mathcal{Z}_U} D_{c(U)}$. If $D_0(\mathcal{Z}_U) > 0$, the survival probability conditional on commitment U is $\lambda^{128} D_{c(U)}(\mathcal{Z}_U)/D_0(\mathcal{Z}_U)$. When this is positive, the surviving response has law $D_{c(U)}$ conditioned on $\mathcal{Z}_U$. Appendix E gives the subsequent oracle and comparison of all outputs and retained quantum state, including aborts, finite sampling, retries and public checks that the response fits in the encoding, under Assumptions F.1 and F.3.

A.2 Finite evaluation of the transport probabilities

This subsection certifies the bound $\mathcal{M}(b) < M$ used above and shows how to enclose the probabilities f and g with finite arithmetic. For the specified dyadic θ, the exact supremum of $\widehat{f} + \widehat{g}$ is $\mathcal{M}(b) = 1 + \vartheta_4(0, \theta^b)/\theta^b$. The key-norm restriction (3) gives $105{,}500 \le b \le 230{,}053$. Set $B = 230{,}053$ and $S = 2^{512}$, and use the exact dyadic base from Appendix D. For each integer $k \ge 0$, compute integers L_k, U_k with $L_k \le S\theta^k \le U_k$ by repeated squaring. For nonnegative intervals $[L, U]/S$ and $[L', U']/S$, use product endpoints $\lfloor LL'/S \rfloor$ and $\lceil UU'/S \rceil$. The base is represented exactly at this scale. Define

$$A_+ = S + 2 \sum_{\substack{1 \le j \le 20 \\ j \text{ even}}} U_{Bj^2} - 2 \sum_{\substack{1 \le j \le 20 \\ j \text{ odd}}} L_{Bj^2},$$

$$A_- = S + 2 \sum_{\substack{1 \le j \le 20 \\ j \text{ even}}} L_{Bj^2} - 2 \sum_{\substack{1 \le j \le 20 \\ j \text{ odd}}} U_{Bj^2} - 2U_{441B}.$$

The omitted alternating tail is between $-2\theta^{441B}$ and zero. Thus $A_-/S \le \vartheta_4(0, \theta^B) \le A_+/S$. Integer arithmetic gives $A_- > 0$, $L_B > 0$, and

$$\frac{1{,}001{,}149{,}955{,}473{,}938}{10^{15}} < 1 + \frac{A_-}{U_B} \le \mathcal{M}(B) \le 1 + \frac{A_+}{L_B} < \frac{1{,}001{,}149{,}955{,}473{,}940}{10^{15}} < \frac{65{,}612}{65{,}536}.$$

In particular,

$$1 < \mathcal{M}(230{,}053) < \frac{65{,}612}{65{,}536}. \tag{A.6}$$

Since $\mathcal{M}$ is increasing, the envelope bound holds throughout the accepted range.

For $u \geq 0$, consecutive partial sums of $T(u, b)$ enclose the exact value, with truncation error after $k = n$ bounded by

$$\theta^{2(n+1)u+(n+1)^2 b}. \tag{A.7}$$

Outward-rounded multiplication by the rational factors in (17)–(18), and addition of the resulting intervals, enclose f and $f + g$. For exact random-bit sampling, increase both the truncation length and arithmetic precision until the interval determined by the random bits read so far lies inside one outcome region. The implementation instead uses bounded-precision arithmetic and has an explicit outcome for steps it cannot decide. Under Assumptions F.1 and F.3, Appendix F incorporates the complete finite-sampler error bound into the signing comparison, including private random bits and the reset of private state after failed attempts. The series bounds in this subsection do not establish those assumptions or that complete finite-sampler bound.

B Measurement Hiding: Background and Proofs

B.1 Fiat–Shamir and the Quantum Random-Oracle Model

A public-coin three-message identification protocol sends a commitment, receives a random challenge, and answers with a response. The Fiat–Shamir transform replaces the challenge by a hash of the public transcript. In the quantum random-oracle model (QROM), the attacker may query this hash in superposition.

The measure-and-reprogram theorem turns a Fiat–Shamir forger making q_H hash queries into an interactive prover whose success is smaller by the factor $(2q_H + 1)^2$, up to the simulation error [13, 14]. A generic reduction cannot avoid essentially this quadratic loss.

A protocol is collapsing if a quantum prover cannot tell whether the verifier measured the accepted response. Don, Fehr, Majenz, and Schaffner [14] call the full-response variant quantum computationally unique responses. Lombardi, Ma, and Spooner also study measurement of a public function of the response [29].

The game in Section 5.2 samples the public key honestly and then allows arbitrary key-dependent adversary state. Liu and Zhandry [28, Definitions 8–9, Lemma 3, and Theorem 8] use the same honest-key setting to prove a weak collapsing result for Lyubashevsky's protocol. Their result retains about one quarter of the unmeasured success probability. We instead bound the distinguishing advantage itself, with an explicit loss, at $q/\|\xi\|_2$ near 2. The adversarial-key collapsing notions of Don, Fehr, Majenz, and Schaffner [14] and of Lombardi, Ma, and Spooner [29] are stronger. Remark B.3 explains why Pretzel needs the honest-key restriction.

B.2 The Integer Secant Check

With the head and residual parameters $t_h = 6$ and $t_r = 2$ of the auxiliary Module-LWE instance, define

$$\Lambda(\xi) = 12 \sum_{i \in z_h} \log_2 \sec\left(\frac{2\pi z_{h,i}}{q}\right) + 4 \sum_{i \in r_e} \log_2 \sec\left(\frac{2\pi r_{e,i}}{q}\right). \tag{51}$$

The verifier scales each contribution λ_i by 2^{20} and rounds upward, which gives the integer table value $q_i = \lceil 2^{20} \lambda_i \rceil$. Rounding upward ensures that an accepted integer sum also bounds the exact real sum. The verifier uses the tables

$$W_h(a) = \left\lceil 2^{20} \cdot 12 \log_2 \sec(2\pi a/q) \right\rceil, \quad 0 \leq a \leq 288,$$

$$W_e(a) = \left\lceil 2^{20} \cdot 4 \log_2 \sec(2\pi a/q) \right\rceil, \quad 0 \le a \le 451,$$

and rejects unless

$$\sum_{i \in z_h} W_h(|z_{h,i}|) + \sum_{i \in r_e} W_e(|r_{e,i}|) \le 30{,}850{,}912. \tag{52}$$

The residual table extends to $a = 451$, the implementation's generic residual cap; the local-label decoder of Section 4.5 rejects $|r_{e,i}| > 450$, so accepted responses never use $W_e(451)$, and the extra entry only enlarges the range on which the inequalities below are checked. Every table entry is a ceiling, so $2^{20}\Lambda(\xi)$ is at most the integer sum in (52), and acceptance gives

$$\Lambda(\xi) \le \frac{30{,}850{,}912}{2^{20}} = 29.421722412109375, \tag{53}$$

which gives $\kappa = 2^{964091/32768}$. Conversely, for any response within the coefficient caps, the integer sum is less than $2^{20}\Lambda(\xi) + 2304$. Both inequalities are deterministic. Since the squared feature norm is $2^{\Lambda(\xi)}$, neither the bound on κ nor the response-norm bound below depends on an estimate of how often the check rejects.

Lemma B.1 (Response norm from the integer secant check)**.** Every accepted Pretzel-(4,4) response satisfies $\|\xi\|_2^2 \le 11{,}350{,}941$. Any two such responses satisfy

$$\|\xi - \xi'\|_2^2 \le 45{,}403{,}764 < 6739^2.$$

Proof. The ceiling tables satisfy the finite scalar inequalities

$$3530a^2 \le 581 W_h(a) + 788 \quad (0 \le a \le 288),$$

$$99{,}019a^2 \le 48{,}804 W_e(a) + 1{,}021{,}315 \quad (0 \le a \le 451).$$

An exact certificate reconstructs all 741 ceiling values with outward rational intervals and checks these integer inequalities. Equality holds at head magnitudes 38, 45 and residual magnitudes 95, 101.

Let S be the accepted integer secant sum and $r = \|r_e\|_2^2$. Multiply the head inequality by 84, add $197{,}501a^2$ to both sides of the residual inequality, and sum over the 1280 head and 1024 residual coordinates. This gives

$$\begin{aligned} 296{,}520\|\xi\|_2^2 \le\ & 48{,}804S + 197{,}501r \\ & + 84 \cdot 1280 \cdot 788 + 1024 \cdot 1{,}021{,}315. \end{aligned}$$

Substituting $S \le 30{,}850{,}912$ and $r \le 3068^2$ yields

$$\|\xi\|_2^2 \le \frac{20{,}034{,}411{,}394}{1765} = 11{,}350{,}941 + \frac{529}{1765}.$$

The squared norm is an integer. The pair bound follows from $\|\xi - \xi'\|_2^2 \le 2\|\xi\|_2^2 + 2\|\xi'\|_2^2$. The proof uses only public acceptance checks. □

B.3 Proof of the Measurement Hiding Theorem

B.3.1 The feature vectors

For one coordinate with $a = \pi x/q$, $b = \pi x'/q$, and weight $t \in \{\eta_h, \eta_r\}$, the feature vectors in (38, 39) are

$$\alpha_x = \left(\sec^{2t} a \sqrt{\tbinom{2t+j-1}{j}} \tan^j a \right)_{j \ge 0}, \qquad \beta_{x'} = \left(\sec^{2t} b \, (-1)^j \sqrt{\tbinom{2t+j-1}{j}} \tan^j b \right)_{j \ge 0}.$$

Since $|\tan a \tan b| < 1$, the binomial series $(1+z)^{-2t} = \sum_{j\ge 0}(-1)^j \binom{2t+j-1}{j} z^j$ converges at $z = \tan a \tan b$ and gives $\langle \alpha_x, \beta_{x'} \rangle = \sec^{2t} a \, \sec^{2t} b \, (1 + \tan a \tan b)^{-2t}$, which is the first identity in (38, 39). The norm is

$$\|\alpha_x\|^2 = \sec^{4t} a \sum_{j\ge 0} \binom{2t+j-1}{j} \tan^{2j} a = \sec^{4t} a \, (1 - \tan^2 a)^{-2t} = \sec^{2t}(2a),$$

because $\cos^2 a \, (1 - \tan^2 a) = \cos 2a$. The same computation gives $\|\beta_x\|^2$, the second identity in (38, 39). Tensor products over the coordinates give (40), and the secant sum (31) is $\log_2 \|\alpha_\xi\|^2$. These feature vectors have infinitely many components. Appendix B.4 approximates them with finite registers, prepares the resulting states with their signed amplitudes, and bounds the truncation error.

B.3.2 Proof of the interference identity

of Lemma 5.5. After tracing out the feature register, the control's $|0\rangle\langle 1|$ block is

$$\frac{1}{2} \sum_{\xi,\xi'} |\xi\rangle\langle\xi'| \otimes \sigma_{\xi,\xi'} \langle B_{\xi'} | A_\xi \rangle = \frac{1}{2\kappa} (K \odot \sigma).$$

The $|1\rangle\langle 0|$ block is its adjoint. Taking the expectation of $X \otimes M$ sums these two blocks and gives (44). The calculation does not act on the side-information operators $\sigma_{\xi,\xi'}$, so it also holds for entangled side information and subnormalized accepted states. □

B.3.3 Proof of Theorem 5.4

Proof. Keep the classical data, including the public matrix, in a register T. Let σ be the subnormalized accepted state, that is, not divided by the acceptance probability:

$$\sigma = \sum_\tau |\tau\rangle\langle\tau| \otimes \sum_{\xi,\xi'} |\xi\rangle\langle\xi'| \otimes \sigma^\tau_{\xi,\xi'},$$

whose trace is the acceptance probability. By the acceptance test, within each classical block all nonzero response entries solve the same equation $H\xi = y_\tau$, with $\|\xi\|_\infty < q/4$ and $\Lambda(\xi) \le \log_2 \kappa$. Let Δ measure only the response register, in the computational basis. Let E be the effect of $\mathcal{B}_2$ for output one; it acts on T, the response, and all side information. Define the signed gap

$$g = \operatorname{Tr} E(\sigma - \Delta\sigma).$$

Both games output zero on rejection, so g is the difference of their unconditional output probabilities. The two accepted states have the same trace, including when that trace is zero. For $M = 2E - I$,

$$\operatorname{Tr} M(\sigma - \Delta\sigma) = 2g. \tag{93}$$

We build a single auxiliary Module-LWE distinguisher for this state. On input (A, w) it gives $A_h = A^\dagger$ to $\mathcal{B}_1$ and runs the acceptance test, outputting zero on rejection. Otherwise it samples u, applies the phase indexed by $(Au - w, u)$ in every classical block, and runs Lemma 5.5 with $X \otimes M$. The acceptance test does not use w, so its rejection probability is the same whether w is a Module-LWE sample or uniform. A real Module-LWE sample gives $\mathsf{T}(\sigma)$ by the dephasing calculation in Section 5.2, and a uniform sample gives $\Delta(\sigma)$. Both maps act blockwise, so by linearity across the classical blocks, Lemma 5.5 gives the gap between the two expectations, namely $2g/\kappa$. Converting the $\{-1, +1\}$ observable to the effect for output one, $(I + X \otimes M)/2$, divides the signed gap by two. The distinguisher's advantage is therefore $|g|/\kappa$. Because the same readout is used for all fixed classical data, no relabeling depending on those data of outputs is needed.

This final two-outcome measurement is implemented directly. Measure the control in the X basis, obtaining $x \in \{-1, +1\}$, run $\mathcal{B}_2$ on the response and its side information, and let $b \in \{0, 1\}$ be its output. Output one exactly when $x(2b - 1) = +1$. The two measurements act on separate registers, so the signed expectation is $\mathbb{E}[x(2b-1)] = \mathrm{Tr}[(X \otimes (2E - I))\rho]$, where ρ is the state after feature preparation. This works for any effect $0 \le E \le I$; it does not require $2E - I$ to be unitary on the response space. A larger reduction that needs a coherent version of this step should keep the purification workspace and output registers of $\mathcal{B}_2$, rather than assume a clean unitary implementation of $2E - I$.

Replacing the ideal feature circuit, phase circuit, and uniform mask by their finite implementations changes the distinguisher's output probability in each of the two experiments by at most $\varepsilon_{\mathrm{iso}} + \varepsilon_\phi + \delta_u$. Hence

$$|g| \le \kappa \, \mathrm{Adv}_{\mathsf{MLWE},\mathrm{aux}} + 2\kappa(\varepsilon_{\mathrm{iso}} + \varepsilon_\phi + \delta_u),$$

which is (32). □

B.4 Details of the Measurement Hiding Proof

B.4.1 Finite feature states

We approximate the coordinate factorization of Section 5.2. For coordinate values x, y, put $a = \pi x/q$ and $b = \pi y/q$, and let t be the noise parameter for that coordinate. One register per coefficient holds the degree K of a term in the binomial series

$$(1 + uv)^{-2t} = \sum_{K \ge 0} (-1)^K \binom{2t + K - 1}{K} (uv)^K. \tag{56}$$

Keeping the signs of the amplitudes, one coordinate may use

$$\widetilde{\alpha}_x(K) = \sec^{2t}(a) \sqrt{\binom{2t + K - 1}{K}} \tan^K(a),$$

$$\widetilde{\beta}_y(K) = \sec^{2t}(b) \sqrt{\binom{2t + K - 1}{K}} (-\tan b)^K. \tag{57}$$

This gives the same inverse kernel and the same norm

$$\langle \widetilde{\alpha}_x, \widetilde{\beta}_y \rangle = \cos^{-2t}(a - b),$$

$$\|\widetilde{\alpha}_x\|^2 = \sec^{4t}(a) \sum_{K \ge 0} \binom{2t + K - 1}{K} \tan^{2K}(a) = \sec^{2t}(2a). \tag{58}$$

These are the feature vectors of Appendix B.3, so neither the verifier's bound nor κ changes. After normalization, the squared amplitudes give K the negative binomial distribution

$$\Pr[K] = \binom{2t + K - 1}{K} (1 - r)^{2t} r^K, \qquad r = \tan^2 a. \tag{59}$$

These probabilities depend only on $|x|$, so the signs are applied separately as phases. The α branch applies the phase $\mathrm{sgn}(x)^K$ for $x \ne 0$, and the β branch applies the additional phase $(-1)^K$. Without the input-sign phase, $\tan(a)$ would be replaced by $|\tan(a)|$, and (57) would fail for negative coefficients. For $x = 0$ only degree zero has nonzero amplitude. A finite register must cut off the degree, and the cutoff must grow with the requested precision. The following bound gives an explicit cutoff, both for Pretzel-(4,4) and for a scalable family of parameters.

A uniform truncation bound. Let d be the total number of response coefficients and suppose $r_i = \tan^2(\pi x_i/q) \le \bar{r} < 1$ on the accepted domain, with $t_i \le t_{\max}$. For a negative binomial variable in (59), its generating function is

$$\mathbb{E}[s^K] = \left(\frac{1-r_i}{1-r_i s}\right)^{2t_i} \qquad (1 < s < 1/r_i).$$

Choose $s = (1+\bar{r})/(2\bar{r})$ when $\bar{r} > 0$. The ratio inside the power increases with r_i, so it is at most two. Markov's inequality therefore gives

$$\Pr[K > D] \le 2^{2t_{\max}} s^{-(D+1)}.$$

If $\bar{r} = 0$, every coordinate has degree zero and no truncation is needed. Otherwise, by a union bound over the d coordinates, the product distribution has omitted mass at most $p_{\text{tail}} = d\,2^{2t_{\max}} s^{-(D+1)}$. Projecting its amplitude state onto the retained degrees and normalizing changes that unit vector by at most $\sqrt{2p_{\text{tail}}}$. Indeed, its overlap with the original unit vector is $\sqrt{1-p}$, where $p \le p_{\text{tail}}$, and $2 - 2\sqrt{1-p} \le 2p$. Since $\log s \ge (1-\bar{r})/(1+\bar{r}) \ge (1-\bar{r})/2$, a cutoff polynomial in $t_{\max}$, $\log d$, $\log(1/\epsilon)$, and $(1-\bar{r})^{-1}$ achieves any prescribed state-vector error ϵ.

For Pretzel-(4,4), the coefficient caps give $|x_i| \le 450 < q/16$, and $\tan(\pi/16) < 1/4$, so one may take $\bar{r} = 1/16$. Using $s = 2$ instead gives

$$\mathbb{E}[2^K] \le (15/14)^{12} < 4, \qquad p_{\text{tail}} \le d\,2^{1-D}.$$

Here $d = 2304 < 2^{12}$. The resulting state-vector error is at most $2^{(14-D)/2}$. Consequently $D \ge 2B + 18$ gives truncation error at most $2^{-B}/4$, leaving the rest of the error budget for normalization, arithmetic, and gate approximation.

Uniform state preparation and its cost. For each coefficient, compute the retained weights in (59) and their partial sums with directed rounding. A sequence of conditional two-outcome rotations then prepares the normalized finite distribution. There are $D+1$ weights per coordinate, so forming the sums and rotations takes a number of arithmetic operations polynomial in d and D. The functions involved are evaluated on the compact domain $r_i \le \bar{r} < 1$, and the required precision is polynomial in $\log q$, $t_{\max}$, D, and the target precision. Branches of zero weight continue with a fixed state. Branches whose total mass is below an allotted absolute-error threshold may also continue with a fixed state, with their omitted amplitude charged to the error budget. This avoids needing relative-error guarantees for arbitrarily small conditional probabilities.

Compute the rotation angles reversibly, apply the controlled rotations, and uncompute the arithmetic workspace. Then apply the input-sign and branch-sign phases described above. The attenuation construction in the next subsubsection supplies the feature norm and the completion states, which are orthogonal between the two branches. By telescoping, the operator-norm errors of the finitely many controlled operations add, so each operation can receive an inverse-polynomial share of the requested error. The coefficient register is retained, so the isometry acts separately on each coefficient string. A uniform error bound for every coefficient string therefore bounds the error of the controlled isometry on arbitrary superpositions and on inputs entangled with side information.

For Pretzel-(4,4) responses, the coefficient caps, feature exponents, and secant threshold used in Appendix B.5 give an explicit choice of cutoff and precision. Given $B \ge 0$, take

$$D = 2B + 20, \qquad R = 2(2304(D+1) + 25), \qquad p = B + \lceil \log_2(32R) \rceil.$$

For exponent $m = 2t \in \{4, 12\}$, condition the degree law on $K \le D$. Its conditional probability of $K = k$ given $K \ge k$ can be evaluated without cancellation as

$$h_k = \frac{\binom{m+k-1}{k}}{\sum_{j=k}^{D} \binom{m+j-1}{j} r^{j-k}}, \qquad k < D.$$

The denominator is at least one, including at $r = 0$, so the computation never divides by an exponentially small tail probability. Its integer coefficients have bit length polynomial in $m + D$. Polynomial evaluation with directed rounding to $O(p + (m + D)\log(m + D + 1))$ bits gives absolute error exponentially small in p. A constant multiple of this precision also handles the square-root behavior of the rotation angles near the endpoints. The $r = 0$ state is prepared exactly.

Allow each rotation block controlled by a table lookup an angle error of at most 2^{-p} and a gate-synthesis error of at most 2^{-p}, counting all of its elementary gates. The count R covers, in both branches, the degree rotations, one attenuation correction per coordinate, and the 25 attenuations for the binary digits of the threshold gap $T - S$ (defined below). Their total operator error is at most $2R2^{-p} \le 2^{-B-4}$. With $d = 2304$, the cutoff gives truncation error at most 2^{-B-3}. Exact Boolean sign phases add no error. The feature register holds a fixed state on the orthogonal completion branches, so that truncation changes only the wanted feature direction. Together these choices give isometry error below 2^{-B}. Table lookups that scan all 741 entries of the tables of Appendix B.2, without quantum random-access memory (QRAM), cost $O(2304 \cdot 741Dp)$ Boolean operations, plus the cost of generating the tables and synthesizing the gates. Table generation evaluates the displayed finite expressions at the requested precision, and its cost is counted; the tables are not free advice. This is a circuit construction with an asymptotic cost bound, not an implementation or a concrete gate count.

To convert this isometry bound into the output-probability error $\varepsilon_{\mathrm{iso}}$ of Theorem 5.4, let V and $\widetilde{V}$ be the ideal and implemented controlled isometries on the accepted domain, with $\|V - \widetilde{V}\|_\infty \le \epsilon$. For any accepted state ρ of trace $a \le 1$, including arbitrary side information, the triangle inequality gives

$$\|V\rho V^\dagger - \widetilde{V}\rho\widetilde{V}^\dagger\|_1 \le 2a\epsilon.$$

Both outputs have trace a, so their difference has trace zero. Any subsequent output effect therefore changes the output probability by at most half this trace norm, namely $a\epsilon \le \epsilon$. The rejection branch is the same in both and contributes no difference. Thus the above choice of parameters gives $\varepsilon_{\mathrm{iso}} \le 2^{-B}$ in each of the real and uniform experiments. This bound needs no conditioning on acceptance and loses no extra factor of two.

For a family of parameter sets, this construction has polynomial cost when $d, t_{\max}, \log q$, $(1 - \bar{r})^{-1}$, and the requested number of precision bits are polynomial in the family parameter. When κ is polynomial, the secant bound keeps each r_i suitably far from 1. Indeed, each nonnegative summand of Λ is at most $\log_2 \kappa$, hence

$$\cos(2\pi x_i/q) \ge \kappa^{-1/(2t_i)}, \qquad 1 - r_i = \frac{2\cos(2\pi x_i/q)}{1 + \cos(2\pi x_i/q)} \ge \kappa^{-1/2}.$$

For a uniform family, a known polynomial upper bound on κ then gives a rational choice of $\bar{r}$. The asymptotic theorem also requires a uniform, efficient public acceptance predicate. For Pretzel, the integer secant test is such a predicate; the exact real-valued secant inequality need not be decided.

B.4.2 Normalizing the feature states

The negative binomial state gives the direction of the feature vector. The interference test also needs its length, scaled as $\|\alpha_\xi\|/\sqrt{\kappa}$. The verifier's integer secant table bounds this length, and the rounding deficits below give it exactly. For coefficient x_i, let

$$\lambda_i = 2t_i \log_2 \sec\left(\frac{2\pi x_i}{q}\right), \qquad q_i = \left\lceil 2^{20}\lambda_i \right\rceil. \tag{67}$$

For a full accepted response put $\Lambda = \sum_i \lambda_i$ and $S = \sum_i q_i$, let T be the verifier's threshold, $T = 30{,}850{,}912$ in units of 2^{-20}, and define $\kappa = 2^{T/2^{20}}$. Then

$$\frac{\|\alpha_\xi\|}{\sqrt{\kappa}} = 2^{-(T-S)/2^{21}} \prod_i 2^{-(q_i/2^{20} - \lambda_i)/2}. \tag{68}$$

Every factor lies in $[0, 1]$, and (68) is exact. Acceptance gives $S \le T$. Since $T = 30{,}850{,}912 < 2^{25}$, the first factor is a product of at most 25 fixed attenuations $2^{-2^{k-21}}$, one for each position k at which the binary expansion of $T - S$ has a one. Each rounding deficit $d_i = q_i/2^{20} - \lambda_i \ge 0$ of the verifier's integer table value q_i contributes one correction rotation of amplitude $2^{-d_i/2}$.

To complete the states to unit length, each attenuation uses a flag qutrit with states $|0\rangle, |A\rangle, |B\rangle$, encoded as $|00\rangle, |10\rangle, |01\rangle$. The circuit for the A branch rotates only the first qubit, and the circuit for the B branch only the second. The all-zero flag string carries the amplitude (68). An A-branch flag string uses only $0, A$, whereas a B-branch flag string uses only $0, B$. Their inner product is therefore zero unless both strings are all zero, even when the input labels ξ and ξ' differ. Consequently only the wanted feature components contribute to $\langle B_{\xi'}|A_\xi\rangle = K(\xi, \xi')/\kappa$ in the ideal construction.

The ideal reference for the error bound ε_{iso} is this same flag construction with exact attenuations and untruncated feature directions. Its completion vectors may depend on the response, within two fixed, mutually orthogonal completion spaces; they need not equal the fixed completion states in Section 5.2. Both choices give the same cross inner products and hence the same interference identity. The feature register is prepared only on the all-zero flag branch and holds a fixed state on every completion branch. Truncating the feature direction therefore changes the full state by the direction error times $\|\alpha_\xi\|/\sqrt{\kappa} \le 1$, and leaves the ideal completion amplitudes unchanged. The finite rotation and truncation bounds above compare the implementation with this reference.

B.5 Measurement Hiding for Pretzel Signatures

Remark B.2 (Reversible decoding). Canonical encoding is a bijection on accepted signatures (Lemma 4.6), and an accepted response determines its commitment labels. Hence decoding can be done reversibly, also with the commitment held in superposition, without leaving a copy of the signature, and accepting a signature is the same test as accepting its response. The measurement-hiding argument therefore works directly with response vectors.

B.5.1 From signature bytes to response vectors

Fix the parameter set, canonical public key, message prehash, salt, commitment labels U, and a supplied challenge c. The map described here, which we call the *lift*, evaluates no hash. It decodes the signature, checks that it is canonically encoded and passes all public verification checks, and maps it to the reconstructed response $\xi = (z_h, r_e)$, where $r_e = \text{center}_q(C(U) - A_h z_h - \delta)$. Every accepted lift satisfies

$$H\xi = y \pmod q, \qquad y = C(U) - \delta. \tag{B.1}$$

The lift is injective on accepted inputs. It keeps z_h, and for fixed U the deterministic canonical encoder recovers the original signature bytes. Lemma 4.6 proves that the response-head and local-label fields are canonical. Injectivity holds on the set of accepted signatures, not on all bounded vectors. Inputs that fail the public checks are rejected before any table is indexed.

By Remark B.2, and since the response also determines the supplied challenge $c = \text{parity}(z_0)$, the lift gives identical complete byte and response measurement operations, including rejection, with fixed public parameters, coherent U, c controls and arbitrary reference systems retained. The same lift, with the 1908-byte encoding, applies to Pretzel-(5,5); the numerical bound below is for Pretzel-(4,4).

The lift also turns two distinct accepted signatures into a short kernel vector. Take two accepted pairs of challenge and signature for the same U. If their lifts are equal, then their challenges, which are the parities of z_0, are equal, and canonical inversion makes their signature bytes equal. Distinct pairs therefore give $d = \xi - \xi' \ne 0$ and $Hd = 0$. Lemma B.1 gives $\|d\|_2 < 6739$, and the coefficient bounds give $\|d\|_\infty \le 900 < q$,

so reduction modulo q cannot turn this nonzero integer vector into zero. Each programmed point is lifted with its own recorded challenge.

B.5.2 Applying the short-preimage theorem

For uniform $A_h \in R_q^{4\times 5}$, the accepted lift satisfies the coefficient and secant bounds of Theorem 5.4, with $\kappa = 2^{964091/32768}$. The auxiliary instance has CBD_2^4 secret and CBD_6^5 error, so the characteristic-function powers are 12 on the head and 4 on the residual. The theorem therefore bounds the signed measurement gap g, including all retained classical records and side information, by

$$|g| \le \kappa a_{\mathrm{aux}} + 2\kappa(\varepsilon_{\mathrm{iso}} + \varepsilon_\phi + \delta_u).$$

Besides the feature-circuit error $\varepsilon_{\mathrm{iso}}$, this includes the errors ε_ϕ of the finite phase circuit and δ_u of sampling the uniform auxiliary coins. Section F.5 specifies the precision of these circuits for these dimensions.

B.5.3 Honest keys and queries made before key generation

The comparison of real and simulated signing of Appendix E ends in a hybrid, the *public endpoint*, computed by a total bounded circuit $\mathsf{Pub}(s_*, A_*, b_*; \mathrm{coins})$ with independent coins and defined for every canonical column b_*, including a uniform one. It produces the setup results and failures, declared statistics, aliases (handles from different setups that return equal keys), saved classical data, caches, the response distribution including aborts, and the workspace of the response circuit and its adjoint. The key-norm restriction (3) is decided by an independent copy of the key search whose candidates are never used as the key and, for Pretzel-(4,4) and Pretzel-(5,5), the search for the first dither whose bounded expansion succeeds runs on the supplied public core key, with the respective expansion bounds of Appendix D.2 and Section I.1. It is total on every canonical b_* and retains exhaustion, so none of these outputs uses the secret and error behind b_*. At rank four, no test of G from Lemma D.3 is made: conditioning an external matrix on G would change the required uniform law. This follows from the identities of that appendix for all saved signing data and this second key search.

The adversary may call key generation at most K times. We call the jth possible call *potential setup* j; it may fail or never be reached. Fix the independent oracle tables other than the challenge oracle, and draw the coin tapes of all K potential setups in advance. Setup makes no challenge query, so each potential setup has a fixed public result or failure. Cache these results without revealing future keys to the adversary. Let $\omega = (P, B, \mu, \rho, U)$ be the measured classical first output, which fixes the typed challenge input and contains the key identity (P, B). Define

$$I(\omega) = \min\{j : \text{potential setup } j \text{ succeeds with identity } (P, B)\},$$

or a null marker if the set is empty. This index is a deterministic function of the fixed classical data and the challenge-free cache; computing it does not measure a coherent register that depends on the response. The first output may come from a hash query made before the matching key was generated, and it is not rejected for that reason. When a response is accepted, however, we require that a successful setup with a matching handle has actually been returned in that branch. The indices count failed setups, so a branch that reaches a later setup with an equal key has also reached every earlier one. Hence all branches agree on the least matching index. The freshness check of a forgery still searches the signing log under every matching handle.

Keep ω and $I(\omega)$ outside the response circuit U_c for challenge c and outside its adjoint. All these circuits act on one fixed workspace, which includes the retained coins and the registers that record the reprogrammed challenge values (the list of input–value replacements that determines the current version of the challenge oracle; Appendix E), and they extend to invalid encodings by total reversible maps. The acceptance projector

checks the supplied challenge, the canonical encoding, that the reconstructed typed point equals the recorded one, that a matching handle was actually returned, and all public bounds. The lift compares the supplied challenge with the retained parities of z_0; it does not query the oracle for a replacement challenge. Thus $P_c = U_c^\dagger \Pi_c U_c$ is a projector on that same workspace. Invalid or unmatched branches are rejected before the response measurement in which the two games differ, and no additional coherent measurement of key ownership changes the game without that measurement.

At the public endpoint, setup and signing coins no longer require the secret, and the comparison of real and simulated signing has already paid for the key-norm restriction and for differences between the implementation and the ideal sampler. For each fixed potential setup and each value of the measurement bit, replace its CBD key sample, which is not conditioned on (3), by a uniform public column. Each replacement costs a key-MLWE distinguishing advantage. The two values of the measurement bit need separate testers, with advantages $a_{\mathrm{key},0}$ and $a_{\mathrm{key},1}$, and the triangle inequality combines them. With a uniform column, the game is exactly the one analyzed above. Thus the measurement gap with honestly generated keys is at most

$$K_* \big[a_{\mathrm{key},0} + a_{\mathrm{key},1} + \kappa a_{\mathrm{aux}} + 2\kappa(\varepsilon_{\mathrm{iso}} + \varepsilon_\phi + \delta_u) \big],$$

where $K_* = K/\lambda_K$ is the setup-guessing factor derived at the end of this subsection, as in Theorem 5.6. For a single potential setup the bracket bounds the gap of that setup. These two hybrids need no further conditioning on the key-norm restriction, no division by the probability of a successful signature, and no normalization over retries.

Hash queries made before key generation do not prevent embedding the external matrix. Draw in advance the matrix seed s_* of the target potential setup and the supplied uniform matrix A_*. Define the whole matrix oracle table from the start by setting $F(s_*) = A_*$ and sampling every other entry independently. Averaged over uniform A_*, this has exactly the same joint distribution as sampling s_* and a random table, then setting $A_* = F(s_*)$. Every coherent query made during preprocessing therefore sees the same, correctly distributed table. Colliding seeds receive the same value, and potential setups that are never reached require no reset. The adapter for raw-stream queries keeps the parameter set's accept/reject pattern and the conditional distribution of the raw stream. This is an identity of the joint distribution of the matrix and raw streams; it does not assume that they are independent. Sampling the public column conditional on the supplied matrix gives exactly the real or the uniform key-MLWE game. In the auxiliary hybrid, the supplied uniform 5-by-4 adjoint is split into the public column and the matrix block before the same construction is applied. It is essential that no external instance is conditioned on the partition by private-stream keys and candidate seeds used in the comparison of real and simulated signing.

Finally, let g_j be the signed measurement gap restricted to classical outcomes eligible for potential setup j. Eligibility is checked before the response measurement, so unreached, failed, and invalid classical outcomes contribute zero in both games. The tester selects j uniformly; the selection succeeds with probability λ_K, independently of the experiment, and aborts with probability $1-\lambda_K$. This gives the signed gap $(\lambda_K/K)\sum_j g_j$. The continuation may test any event, in particular that both responses accept and $c \neq d$. The testers are constructed for this full signed sum before absolute values are taken. The factor $K_* = K/\lambda_K$ is the cost of guessing the setup; no tester is run repeatedly. Appendix F combines this measurement comparison with the off-diagonal projector bound.

B.6 Why Honest Key Generation Matters

Remark B.3 (Measurement is visible for an adversarially chosen key). An adversary that chooses the key can know a nonzero short vector s with $Hs = 0$. Given an accepted response ξ_0, the vector $\xi_1 = \xi_0 + 2X^j s$ satisfies the same public equation and has the same leading parities. If both responses pass all public checks, their equal superposition distinguishes measurement from no measurement with advantage $1/2$. Knowing s is not enough to show that such a pair can be found efficiently for every key and every challenge received after

the first message, so the proof below constructs one explicitly. It uses the key $b = 0$, which the public-key format permits, with witness $s = (1, 0, 0)$, and a single first message that works for every challenge. For any fixed public dither whose quantization residual has norm at most 3000, both responses pass the Pretzel-(4,4) verifier V_{ID} of Definition 4.5, which takes the challenge as input instead of hashing, and the distinguishing advantage is $1/2$. If the dither is uniform, this condition holds with probability greater than $1 - 2^{-370}$. The example uses an adversarially chosen key and says nothing about keys produced by honest key generation. Its probability bound assumes a uniform dither. For a seed produced by honest setup, the first successful bounded expansion gives that law in the ideal independent-stream model of Appendix D.2. A chosen key with $b = 0$ is still outside the honest key experiment. The spread condition of Lemma D.3 concerns random response rows and does not prevent this adversarial construction.

of Remark B.3. First suppose that, for a fixed first message and the challenge c received after it, the prover can efficiently construct two accepted responses ξ_0 and $\xi_1 = \xi_0 + 2X^j s$, where $s \neq 0$ and $Hs = 0$, so $\xi_1 \neq \xi_0$. The shift preserves the public equation and the leading parities, and Lemma 4.6 gives two distinct canonical byte strings. The prover prepares their equal superposition. In its final stage $\mathcal{B}_2$, which receives the response after the possible measurement, it measures in a basis containing the sum and difference of these two byte strings. The outcome for the sum has probability one without the response measurement and probability $1/2$ after it. So, conditioned on obtaining such an accepted pair, the advantage is $1/2$.

For the key $b = 0$, the witness is $s = (1, 0^\ell, 0^m)$, and the verifier's bounds can be checked in closed form. Choose any canonical matrix seed and set $b = 0$, so $A_h = [0 \mid A_0]$. Fix the public dither δ and choose a salt ρ. Put $U = Q(\delta)$ and $y = C(U) - \delta$. For every challenge $c \in C_{255}$ define

$$\xi_0 = ((c, 0^\ell), y), \qquad \xi_1 = ((c + 2X^0, 0^\ell), y).$$

Fix a message prehash μ and send the first message (ρ, U) before receiving c. All checks below are those of $V_{\mathrm{ID}}(\mathsf{pk}, \mu, \rho, U, c; Z)$ on the canonical encoding Z of the response. For ordinary signature verification, take c to be the folded hash of the fixed public key, message, salt, and labels U. The same construction then also satisfies the final hash comparison. The public key is chosen by the adversary; its block $A_h = [0 \mid A_0]$ is not uniform, so this is not the experiment of Definition 5.3. The two responses are distinct but have the same parity vector c. They solve the same public equation because A_h maps both heads to zero, and their local commitment-label indices are equal. We now check that both responses pass verification with the given challenge. The larger head has coefficients of magnitude at most 3 and squared norm at most 136. The signed Rice images of these coefficients are at most 6, so all leading quotients are zero. All coefficients of z_s are zero, so the total head quotient sum is zero. For both responses the public context $A_h z_h + \delta$ equals δ. Since $U = Q(\delta)$, every local commitment-label index is zero and the list of label exceptions is empty. Both responses therefore fit in the encoding of the head and local-label fields. Residual coefficients have magnitude at most 113. If $\|y\|_2 \leq 3000$, the residual norm bound holds. Also $|2\pi y_i/q| < 1/10$ and $\log_2 \sec u \leq u^2$ for $|u| \leq 1/10$, so

$$\Lambda(\xi_1) \leq (2\pi/q)^2(12 \cdot 136 + 4 \cdot 3000^2) < 26.$$

Rounding the table entries adds at most $2304/2^{20}$ to this sum, which stays below the verifier's threshold in (53). The smaller head satisfies the same bounds.

Now suppose the dither is uniform, so its 1024 coordinates are independent and uniform. Quantizer cells have size 225 or 226. Conditioned on the cell, the squared centered error has mean at most 4256.5 and lies in $[0, 113^2]$. Hoeffding's inequality therefore gives

$$\Pr[\|y\|_2 > 3000] \leq \exp\left(-\frac{2(9{,}000{,}000 - 1024 \cdot 4256.5)^2}{1024 \cdot 113^4}\right) < e^{-257} < 2^{-370}.$$

The prover checks the residual norm before sending the first message and falls back to a strategy with zero advantage if the check fails. The same first message works for every challenge received afterward. With a uniform dither, the prover's unconditional advantage is therefore greater than $(1 - 2^{-370})/2$, without resampling. The construction of the accepted pair is deterministic and needs only the residual-norm bound; it makes no independence assumption about a fixed dither. □

C Quantum Extraction Bounds

C.1 The Sequential Extraction Bound

Lemma C.1 (Sharp off-diagonal projector bound)**.** For $C \geq 2$ projectors P_c on a common workspace and a unit vector ψ, put $\gamma = C^{-1} \sum_c \|P_c\psi\|^2$ and

$$T_\perp = C^{-2} \sum_{c \neq d} \|P_d P_c \psi\|^2, \qquad f_C(x) = \frac{x(x - 1/C)_+^2}{1 - 1/C}.$$

Then $T_\perp \geq f_C(\gamma)$; f_C is nondecreasing and convex. The bound is attained for every $\gamma \in [0, 1]$.

Proof. With $S = \sum_c P_c$,

$$L_0 = \sum_{c \neq d} \langle P_d\psi, P_c\psi \rangle = \langle S^2 \rangle - C\gamma \geq C^2\gamma^2 - C\gamma.$$

Cauchy–Schwarz applied to the paired vectors $P_d\psi$ and $P_d P_c\psi$ gives $|L_0|^2 \leq C(C-1)\gamma\, C^2 T_\perp$. For $\gamma > 1/C$, dividing by $C^3(C - 1)\gamma$ proves the claim; otherwise the claimed lower bound is zero. By purification, the same bound holds for mixed states and with arbitrary reference systems. The function f_C is nondecreasing and convex: above $a = 1/C$ its derivatives are $(x - a)(3x - a)/(1 - a)$ and $(6x - 4a)/(1 - a)$, and at a its value and first derivative are zero, matching the zero function below a. By Jensen's inequality, the bound therefore also holds after averaging over classical outcomes. An outcome with subnormalized probability is handled by adding a branch with zero success probability.

The dependence on γ is sharp for arbitrary projector families. For $\gamma \geq 1/C$, take a unit vector e_0 and unit regular-simplex vectors $w_1, \ldots, w_C$ orthogonal to it, with $\langle w_c, w_d \rangle = -1/(C - 1)$ for $c \neq d$. Set $v_c = \sqrt{\gamma}\, e_0 + \sqrt{1 - \gamma}\, w_c$, $P_c = |v_c\rangle\langle v_c|$, and $\psi = e_0$. Then $\|P_c\psi\|^2 = \gamma$ and $\langle v_d, v_c \rangle = (C\gamma - 1)/(C - 1)$ off the diagonal, so $T_\perp = f_C(\gamma)$. For $0 \leq \gamma \leq 1/C$, take orthonormal $v_1, \ldots, v_C, e_0$ and $\psi = \sqrt{\gamma} \sum_c v_c + \sqrt{1 - C\gamma}\, e_0$; the same rank-one projectors give $T_\perp = 0$. Thus any stronger bound that holds for all projector families must use information beyond the average acceptance probability. □

C.2 Bounding Extraction by Filtering Responses

For Pretzel-(4,4), put $q = 7681$, $\kappa = 2^{964091/32768}$, and $\kappa_0 = \sqrt{\kappa}$, using the integer secant bound of Appendix B.2. Throughout this subsection, a distinguishing advantage is the absolute *difference* of two probabilities of outputting one. A SIS advantage is a success probability, where success includes the public checks that the output is nonzero, short, and satisfies the equation. By Lemma B.1, every SIS term here uses radius $\beta = 6739$, both for a pair of responses obtained directly from a forgery and for sequential extraction. The constructed algorithms run without an external random oracle. Their queries are answered by *oracle simulators*, finite circuits that simulate the random oracles (as in Section F.2). The algorithms receive one copy of the original advice state, in the same advice class as the underlying assumption. Their costs include oracle-dependent preprocessing, storage, input preparation, and every forward and inverse operation. We infer no numerical trade-off between running time and advantage from attack estimates.

Lemma C.2 (Filtering accepted responses)**.** Let $A_h \in R_q^{4\times 5}$ be uniform, put $A = A_h^{\dagger}$, and use the auxiliary Module-LWE sample (A, w) with CBD_2^4 secret and CBD_6^5 error. Given A, neither the preparation nor the continuation accesses w. In each fixed classical outcome, let σ be the subnormalized state of accepted Pretzel-(4,4) lifts satisfying the public coefficient and integer secant checks of Appendix B.2, with $[A_h \mid I_4]\xi = y$ for that outcome's fixed target. The state may include arbitrary side information and a reference system. Let $0 \preceq E \preceq I$ be the effect of a continuation, applied in both cases, that does not read a new copy of the response, and set

$$p_0 = \mathrm{Tr}(E\sigma), \qquad p_1 = \mathrm{Tr}(E\Delta_\xi(\sigma)).$$

For the actual coefficient and secant bounds, the finite auxiliary tester constructed in the proof, with complete work W_+, has an advantage a_+ that satisfies

$$p_0 \le \sqrt{\kappa}\,(p_1 + a_+ + e_R + e_U). \tag{C.1}$$

Here e_R and e_U bound the errors in the tester's output probabilities, with a real and with a uniform sample, relative to the ideal phase and filter. If the tester's finite oracle simulator changes either output probability by at most $\delta_{\mathrm{sim},+}$, replace a_+ in (C.1) by $a_+ + 2\delta_{\mathrm{sim},+}$. The probabilities average over the uniform public matrix and all classical outcomes, without normalizing the accepted mass.

Proof. Write $z_i = \pi\xi_i/q$, with $m_i = 12$ on the 1280 head coordinates and $m_i = 4$ on the 1024 residual coordinates. With a real sample, the auxiliary phase multiplies the state entrywise by the real kernel $\Phi(\xi, \xi') = \prod_i \cos^{m_i}(z_i - z_i')$; with a uniform sample, it is exactly Δ_ξ. These are the identities of Section 5.2. Indeed, the phase vector is $(Au - w, u) = H^{\dagger}(u - s) + (-e, s)$ for $H = [A_h \mid I_4]$. The first term contributes only a phase common to all solutions of the equation fixed by the classical outcome, and averaging over the independent short noise gives the kernel.

With $\kappa_0 = \sqrt{\kappa}$, apply a filter whose success outcome has the diagonal Kraus operator

$$D_a = \mathrm{diag}(a(\xi)), \qquad a(\xi) = \kappa_0^{-1/2} \prod_i \sec^{m_i} z_i.$$

Because $|z_i| < \pi/4$ and $\cos^4 z - \cos(2z) = \sin^4 z \ge 0$, the secant bound on accepted responses gives

$$a(\xi)^2 \le \kappa_0^{-1} \prod_i \sec^{m_i/2}(2z_i) \le 1.$$

Thus D_a is a contraction. Multiplying the kernel by the filter amplitudes gives

$$D_a(\Phi \circ \sigma)D_a = \kappa_0^{-1}(G_+ \circ \sigma), \qquad G_+(\xi, \xi') = \prod_i (1 + \tan z_i \tan z_i')^{m_i}.$$

The finite binomial expansion of G_+ has constant term one, and all its other terms are Gram kernels with nonnegative coefficients. Hence $G_+ - \mathbf{1}\mathbf{1}^* \succeq 0$. The Schur product theorem, applied with the identity on the reference system, gives

$$D_a(\Phi \circ \sigma)D_a \succeq \sigma/\kappa_0, \qquad D_a\Delta_\xi(\sigma)D_a \preceq \Delta_\xi(\sigma).$$

The second inequality holds blockwise, including when lift labels repeat. Let r and u be the probabilities that the implemented tester outputs one with a real and with a uniform sample. Then $p_0/\kappa_0 \le r + e_R$ and $u \le p_1 + e_U$, and $r - u \le |r - u|$ proves the claim. No sign is chosen separately for different classical outcomes. The argument gives only this one-sided bound; it does not bound $|p_0 - p_1|$ with the smaller coefficient. □

Constructive finite filter. Let q_i be the verifier's integer table value for coordinate i's contribution to the secant sum. Put $S = \sum_i q_i$ and $T_0 = 30{,}850{,}912$. On accepted inputs the exact amplitude factors as

$$a(\xi) = 2^{-(T_0-S)/2^{22}} \prod_i \left[2^{-q_i/2^{22}} \sec^{m_i}(\pi \xi_i/q) \right].$$

Every factor is at most one by the preceding cosine inequality. We use 2304 coordinate factors and 25 binary factors for the gap $T_0 - S$. For a target factor $\alpha \in [0,1]$ and an integer $p \geq 1$, compute an enclosing interval of width at most 2^{-p} with lower endpoint α_L, and intersect it with $[0,1]$, which contains α by the bound just proved. Put $K = \lfloor 2^{p-1}(1+\alpha_L) \rfloor$. Prepare p coin qubits with Hadamard gates, apply the phase $+1$ to coin values below K and -1 to the others, cleanly uncompute the comparison, and apply the Hadamard gates again. The amplitude of the all-zero coin state is $b = 2K/2^p - 1$, with $0 \leq b \leq \alpha$ and $\alpha - b \leq 3\,2^{-p}$. Store K in $p+1$ bits, so that the endpoint $K = 2^p$, which gives amplitude exactly one, is representable.

Keep all $2329p$ coins. Neither the continuation nor its tests act on these filter coins. Uncompute the lift circuit and all deterministic lookup scratch before running the continuation, whose final output bit additionally requires every filter coin to be zero. A telescoping product bound gives $\|D_a - D_b\| \leq 3 \cdot 2329\, 2^{-p}$, and thus an error of at most $6 \cdot 2329\, 2^{-p}$ in the event probability. Discarding the coins instead would implement a different channel.

The 741 coordinate constants and 25 gap constants are computable functions of the fixed secant table. Rational series for cosine, logarithm, and exponential with directed rounding give any requested interval width on their compact domains. Denote the complete cost of generating the constants, including remainder bounds, integer arithmetic, and table preparation, by $C_{\mathrm{tab}}(p)$. Linear scans for equality, reversible comparisons, and clean arithmetic give a finite scan cost of $O(2329 \cdot 741(p + \log q))$ gates in the H, S, CNOT, Toffoli basis, without QRAM. Denote the actual charged cost of using the filter by $C_{\mathrm{use}}(p)$; the asymptotic bound does not supply a numerical gate count. The uniform mask and its modular phase have their own complete costs. With b mask bits per coordinate and thirteen binary phase gates whose errors as isometries are at most ν_j, the error e_+ in either case, real or uniform, satisfies

$$e_+ \leq 6 \cdot 2329\, 2^{-p} + \frac{1024 \cdot 7681}{4 \cdot 2^b} + 2 \sum_{j=1}^{13} \nu_j. \tag{C.2}$$

Add $C_{\mathrm{tab}}(p)$, all mask and phase synthesis and adapters, initialized wires, and $C_{\mathrm{use}}(p)$ to the tester's cost. The bound (C.2) can be made arbitrarily small, but it is not set to zero in numerical bounds. The proof for the finite table does not require enumerating the formal monomials of the Gram expansion.

Folded projector index and raw programmed value. Fix the classical values of measure-and-reprogram, and let σ be its entry state, including all advice and retained workspace. Write a uniform raw label as $r = (u, b)$ with $u \in \{0,1\}^{255}$ and $b \in \{0,1\}$, and set $c = \mathrm{Fold}_{255}(u)$. The inverse $u(c)$ is unique. Retain b in a new workspace register B with state $I_B/2$, independent of σ. It is the unused bit of the same uniform 256-bit draw, not a new challenge distribution. Put

$$\widetilde{U}_c = \sum_{b=0}^{1} |b\rangle\langle b|_B \otimes U_{(u(c),b)}, \qquad P_c = \sum_{b=0}^{1} |b\rangle\langle b|_B \otimes U_{(u(c),b)}^{\dagger} \Pi_{(u(c),b)} U_{(u(c),b)}.$$

Each U_r installs the full raw value in the saved input–value pair for the programmed hash and executes its original continuation, including consistent raw queries. Each Π_r is the original acceptance projector for verification with the challenge given instead of hashed. These P_c are projectors on one common workspace, indexed by exactly $C = 2^{255}$ distinct folded challenges. For $\widetilde{\sigma} = (I_B/2) \otimes \sigma$ their average is exactly

$$\frac{1}{C} \sum_c \mathrm{Tr}(P_c \widetilde{\sigma}) = \frac{1}{2C} \sum_{c,b} \mathrm{Tr}(U_{(u(c),b)}^{\dagger} \Pi_{(u(c),b)} U_{(u(c),b)} \sigma) = \gamma.$$

Thus the off-diagonal lemma gives $f_{2^{255}}(\gamma)$ with no same-fold subtraction. In the second response execution choose a new uniform folded challenge d, retain the same b, and use $\widetilde{U}_d$. The adjoint of the first execution preserves B, as does the response measurement. The event $d \neq c$ now forces unequal leading parities and hence unequal canonical lifts. The second execution is an extractor operation; it need not independently redraw all 256 raw bits. It still implements a consistent raw oracle with its chosen full programmed hash value. Mixed advice, rejected branches and arbitrary references are included in the projector statement.

The same construction applies to Pretzel-(5,5), whose fold has the same 255-bit index. For Pretzel-(3,3), use the unique 204-bit challenge rank and a retained uniform 52-bit suffix register instead of B; the raw average has denominator $2^{204}2^{52} = 2^{256}$ and the projector index has size 2^{204}. At the initial measure-and-reprogram draw the whole raw label, including this suffix, is fresh. It is that drawn suffix which is kept for the extractor's second execution. This does not replace the consistent raw-prefix interface by a zero-padded rank. The separate coherent return test still uses $N = 2^{256}$ raw labels; none of its formulas is reindexed by this argument.

Extracting responses to different challenges. The event of interest is that both responses are accepted and their challenges differ. We include this condition directly in the measurement comparison, so every successful outcome with a measured first response yields a nonzero short vector. Apply Lemma C.1, with its function f_C, to the extraction experiment and its projectors on a common workspace, and put the condition $c_2 \neq c_1$ into the continuation E from the outset. With ideal circuits and an unmeasured first response, this gives $p_0 \geq f_C(\gamma)$, including mixed advice, all classical outcomes, and abort branches. When the first response is measured, the continuation uses the same inverse and the same circuit for the second response. It measures the second accepted response only at the end, and E does not read the recorded first response. Different challenges ensure distinct canonical lifts, so a measured success itself yields the vector of Theorem 5.1. No subtraction for equal challenges and no second measurement comparison is needed.

For a uniform relation matrix, let s_+ be the success probability of this SIS algorithm after all its oracles are replaced by finite simulators, and let $\delta_{\mathrm{sim,ext}}$ be the error of that replacement. Lemma C.2 gives

$$f_C(\gamma) \leq B_+ := \kappa_0 \left[s_+ + \delta_{\mathrm{sim,ext}} + a_+ + 2\delta_{\mathrm{sim},+} + e_R + e_U \right]. \tag{C.3}$$

The absolute-gap argument of the measurement-hiding theorem applies independently to the same uniform setting. With its own tester, costs, and errors, it gives

$$f_C(\gamma) \leq B_{\mathrm{abs}} := s_{\mathrm{abs}} + \delta_{\mathrm{sim,ext,abs}} + \kappa \left[a_{\mathrm{abs}} + 2\delta_{\mathrm{sim,abs}} + e_{\mathrm{abs},R} + e_{\mathrm{abs},U} \right].$$

Thus one may use $\min(B_+, B_{\mathrm{abs}})$. In B_+ the SIS term has coefficient $\sqrt{\kappa}$, which cannot be dropped.

Proof of Corollary F.5. Work after the comparison of real and simulated signing and the collision comparison, which are unchanged. The off-diagonal projector argument shows that the left-hand side is a lower bound on the probability p_0^{honest} of the sequential event without measurement. Use the partition by potential setups and the total public circuit of Appendix B.5. The classical first output and its least matching potential setup are kept outside U_c and $U_c^\dagger$. Acceptance, and the check that a matching handle was actually returned, are evaluated before the response is copied in the measured game. Each potential setup is selected with probability $1/K_*$, so the selected setup contributes $p_0^{\mathrm{honest}}/K_*$ to the unmeasured event of one tester.

In that tester, replace only the selected public column by a uniform column. This changes its event probability by at most $a_{\mathrm{key},0,+} + 2\delta_{\mathrm{sim}}$. As in that appendix, embedding the matrix in the oracle table from the start accounts for all queries made before key generation and for the raw-stream adapter. The external instance is not conditioned on any event. Preparing or inverting this public experiment needs no secret key, and the auxiliary column is withheld from the preparation and the continuation. Apply the filtering lemma (Lemma C.2) to this uniform endpoint. After the first lift is measured, the experiment is already a SIS

algorithm for a uniform relation matrix, so its success is at most $s_{\mathrm{ext},+} + \delta_{\mathrm{sim}}$. The auxiliary gap is at most $a_{\mathrm{aux},+} + 2\delta_{\mathrm{sim}}$. Thus

$$\frac{p_0^{\mathrm{honest}}}{K_*} \le a_{\mathrm{key},0,+} + 2\delta_{\mathrm{sim}} + \kappa_0\big(s_{\mathrm{ext},+} + \delta_{\mathrm{sim}} + a_{\mathrm{aux},+} + 2\delta_{\mathrm{sim}} + e_R + e_U\big).$$

Multiplying by K_* proves the corollary. The argument never returns the measured experiment to honestly generated keys. It therefore needs neither a second key comparison at the measured endpoint nor a separate relation-key comparison. These comparisons are avoided by the order of the hybrids, not by a stronger Module-LWE assumption. The unchanged term P_{pair} covers the separate case of a strong forgery at a programmed point. The loss from replacing signing by its public simulation already pays for the key-norm restriction and the finite implementation, so no further term for either, or for retry normalization, is added here. □

A return test for ordinary unforgeability. The return test uses the collapse-binding argument of Unruh [43, Theorem 32] and Dall'Agnol and Spooner [8, Theorem 28], extended to large challenge sets by Jackson, Miller, and Wang [25, Proposition 3]. They bound the measured case by a reduction to Module-LWE under a parameter condition stated in their own notation [25, Proposition 4]. We do not use that reduction: Lemma C.2 bounds the measured case directly for the parameters considered here.

Proof of Corollary F.6. Use g and B_R as defined in Corollary F.6. Recall $N = 2^{256}$ and $Q = |+_N\rangle\langle+_N|_{\mathsf{R}} \otimes I$ from the coherent challenge interface stated before the corollary. First, let $\rho \succeq 0$ be a subnormalized state with $\operatorname{Tr}\rho \le 1$ and $Q\rho Q = \rho$, with Q as above, and let P be an orthogonal projector; for $\operatorname{Tr}\rho = 1$ the following inequality is [8, Proposition 29] and [25, Lemma 3]. If $\operatorname{Tr}\rho = 0$, then $\rho = 0$ and all three expressions below are zero. Otherwise, write $\rho = \sum_i \lambda_i |\varphi_i\rangle\langle\varphi_i|$ with unit vectors φ_i in the range of Q. Cauchy–Schwarz gives $\langle\varphi_i|P|\varphi_i\rangle = \langle\varphi_i|QP|\varphi_i\rangle \le \|QP\varphi_i\|$ and hence

$$\operatorname{Tr}(P\rho)^2 \le \Big(\sum_i \lambda_i\Big) \sum_i \lambda_i \|QP\varphi_i\|^2 = \operatorname{Tr}\rho \ \operatorname{Tr}(QP\rho P).$$

Consequently,

$$\operatorname{Tr}(QP\rho P) \ge \frac{\operatorname{Tr}(P\rho)^2}{\operatorname{Tr}\rho} \ge \operatorname{Tr}(P\rho)^2.$$

For classical outcomes, apply the same weighted Cauchy–Schwarz inequality to all spectral weights in all outcomes together. Their total mass is at most one; outcomes of probability zero contribute zero. This gives the averaged inequality without normalizing accepted mass.

At the measured point, prepare R in $|+_N\rangle$ with Hadamard gates. By the coherent challenge interface, $U = \sum_r |r\rangle\langle r| \otimes U_r$ and $\Pi = \sum_r |r\rangle\langle r| \otimes \Pi_r$, and the state ρ entering U satisfies $Q\rho Q = \rho$. Put $P = U^\dagger \Pi U$. Lemma 1 of Don, Fehr, and Majenz [13] holds for every oracle, every reprogrammed value, and every projection. Block diagonality therefore identifies $\operatorname{tr}(P\rho)$ with the average success γ of the prover run with a classical challenge. In the public-endpoint game, the event counted by ϵ_0 implies that the output point was never programmed, and hence that Π accepts it. All these predicates are diagonal in a common basis, so $\gamma \ge \lambda_{\mathrm{MR}}\epsilon_0/(2q_B+1)^2$. With $\sigma = \Pi U\rho U^\dagger\Pi$ and $E = UQU^\dagger$, the unmeasured return probability $p_0 = \operatorname{Tr}(E\sigma)$ is at least γ^2.

Let Δ_ξ dephase the canonical lift ξ of the response and put $p_1 = \operatorname{Tr}(E\Delta_\xi(\sigma))$. The lift retains z_0, and acceptance requires $\mathrm{parity}(z_0)$ to equal the folded challenge. The fold discards the last raw bit and Fold_{255} is a bijection onto C_{255}, so each accepted lift is compatible with exactly two raw challenges. Since $U^\dagger$ is block diagonal in r, the projection of $|+_N\rangle$ onto such a pair has squared norm $2/N$, and $p_1 \le \operatorname{Tr}(\sigma)\, 2^{-255}$. This holds for every key, matrix, and realization of the oracle simulators.

By (10), every accepted lift satisfies $H\xi = C(U) - \delta \pmod q$ (this is the relation $[A_h \mid I_4]\xi = y$ required by Lemma C.2). The classical first output fixes U, the public key, and the dither, so this target is independent of the raw challenge r. The effect E acts on the response register itself and reads no new copy of it, and the challenge register is admissible side information. Lemma C.2 therefore applies with continuation E.

The key tester selects a potential setup as in the proof of Corollary F.5. Let S be the event that selection succeeds, the least matching potential setup is the selected one, and the response is accepted. The tester measures S. On S it applies $U^\dagger$, measures Q, and outputs one exactly when the challenge register returns. Outside S it outputs one exactly when 240 fresh fair coins are all zero. It needs no secret key and no auxiliary sample. Let X be the probability that S occurs and the challenge register returns, and let Y be the probability of S. The superscript h marks the honest endpoint and u the endpoint whose selected column is uniform. The tester outputs one with probability $X + 2^{-240}(1 - Y)$. Replacing the selected column by a uniform column changes this probability by at most $a_{\mathrm{key},R} + 2\delta_{\mathrm{sim}}$, so

$$X^{\mathrm{h}} - 2^{-240}Y^{\mathrm{h}} \le X^{\mathrm{u}} - 2^{-240}Y^{\mathrm{u}} + a_{\mathrm{key},R} + 2\delta_{\mathrm{sim}}.$$

The check that a matching handle was returned gives every accepted outcome a least matching potential setup, and selection is independent of the experiment. Restricting to each potential setup therefore partitions the accepted mass and the returned mass. Since $\operatorname{Tr}\sigma = \operatorname{tr}(P\rho) = \gamma$, this gives $Y^{\mathrm{h}} = \gamma/K_*$ and $X^{\mathrm{h}} = p_0/K_*$, with p_0 the unmeasured return probability at the honest endpoint.

The auxiliary tester embeds the uniform 5×4 adjoint in the same potential setup and inserts the filter with success operator D_a. The bound on p_1 holds for every key and every classical outcome. At the uniform endpoint the dephased return restricted to S is therefore at most $2^{-255}Y^{\mathrm{u}}$. The lemma, including its clause for the finite oracle simulator, gives

$$X^{\mathrm{u}} \le \kappa_0\bigl(2^{-255}Y^{\mathrm{u}} + a_{\mathrm{aux},R} + 2\delta_{\mathrm{sim}} + e_R + e_U\bigr).$$

Since $\log_2 \kappa_0 = 964{,}091/65{,}536 < 15$ and $Y^{\mathrm{u}} \ge 0$, the right-hand side minus $2^{-240}Y^{\mathrm{u}}$ is at most $\kappa_0(a_{\mathrm{aux},R} + 2\delta_{\mathrm{sim}} + e_R + e_U)$. The two displays, $p_0 \ge \gamma^2$, and multiplication by K_* give $\gamma^2 \le 2^{-240}\gamma + B_R$. The polynomial $x^2 - 2^{-240}x - B_R$ is nonpositive exactly between its roots, and its smaller root is at most zero. Since $0 \le g \le \gamma$, the inequality also holds at g, which proves (F.3). The larger root is at most $2^{-240} + B_R^{1/2}$, which gives the bound on ϵ. No short kernel vector is extracted, so the SIS success term and its oracle-simulation error do not appear. □

Both return testers run the shared prefix once, U once, and $U^\dagger$ at most once. They make at most $2(F_H + F_U) + 2 \le Q_{\mathrm{bin}}$ binary calls and $2F_M \le Q_{\mathrm{mat}}$ matrix calls, and each of U and $U^\dagger$ makes at most q_B challenge queries. The oracle simulators of Section F.2 and their error δ_{sim} are reused unchanged. The register R adds 256 qubits, 512 Hadamard gates, 512 CNOT gates, and 256 terminal measurements, with no oracle call. The key tester also draws 240 fair coins. The auxiliary tester inserts the lift circuit, mask, modular phase, and diagonal filter once, at the costs $C_{\mathrm{tab}}(p)$ and $C_{\mathrm{use}}(p)$ above. Compared with the testers for sequential extraction, each return tester omits the second response pass, its acceptance test, and the second challenge.

Corollary F.6 concerns EUF only. For SUF, a forgery at a programmed point has no fresh measured point, so its term P_{pair}, including the SIS success s_{pair}, is kept. For one fixed target, the unmeasured return over two branches is at most twice the measured return, so this approach cannot replace that term by one with coefficient κ_0. For EUF, the corollary is an alternative to Corollary F.5, and one may take the smaller of the two bounds. Both apply to the same scheme under the same key and auxiliary Module-LWE assumptions.

Complete costs and oracle access. All advantages are evaluated at the costs of the complete constructed algorithms. The cost of the diagonal filter includes table construction and use, all adapters, mask generation,

the modular inner product, finite phase synthesis, retained coins, and the final test that all coins are zero. The absolute-gap tester has its own complete cost. Neither tester simply inherits the forger's running time, and neither treats a finite oracle simulator as a unit-cost oracle.

Acceptance tests use the supplied challenge and do not hash a second one. Once the matrix, the fixed dither, and the accepted response and its associated classical values are retained, the phase and the filter are pure arithmetic. Any oracle access needed to provide these inputs must be charged as a query; a gate budget alone does not provide that access.

Use the bounds $Q_{\mathrm{bin}}, Q_{\mathrm{mat}}$ and widths I, L of Section F.2, which already cover the implementation's interfaces. They also cover the three executions, inverses, and confirmations, so they must not be multiplied by three again. Inserting the filter with success operator D_a adds only its own charged local work and retained registers. The cache of potential setups, the bounded selection of a setup, raw-prefix queries whose inputs coincide with hash inputs used elsewhere in the experiment (raw aliases), and the full history of the public simulator remain charged. The binary sparse Fourier table and the 205-polynomial matrix simulator are rebuilt for these complete budgets, with initialization, reversible Horner evaluation, table updates, storage, and coefficient sampling charged. Their error is the δ_M displayed in that section. Neither the setup-guessing factor K_* nor the coefficient κ_0 means that a hardness tester is run that many times.

All comparisons use one copy of the original advice state. Inverses are literal: they act on the original retained work, including the oracle, signing, and histories of programmed hash values. Filter coins and fresh mask coins are kept separate from that work. The continuation never tests or resets them, and the filter's test that its coins are zero is applied only to the final event. The proof uses no coherent reflection about the unknown advice, no extra copy of the advice, no oracle normalized over retries, and no signer that needs the witness of the external key.

Apply the diagonal filter to the canonical reconstructed response. Accepted responses form a subset of the bounded preimages considered above, so the secant bound and the Schur-kernel inequality apply with all reference registers retained. The unmeasured key tester includes the search for the first dither whose bounded expansion succeeds and its exhaustion outcome on every supplied canonical core. The filter uses only accepted-response bounds, not a row success lower bound or the event G of Lemma D.3. The oracle-simulator budgets keep the setup allowance $7200K$ in R from Section F.2, which exceeds the $5144K$ key-seed, dither and other setup calls including cleanup. The complete tester allowances cover the bounded search and its local work and storage. The finite filter costs and errors are charged separately, as above.

D Byte Formats, Sampling, and Encoding Certificates

This appendix specifies Pretzel-(4,4): matrix expansion, key generation, the public-key layout, the public hash inputs, and the schedule of private randomness used in signing. The key-seed inputs, the input encoding of the private attempt streams, and the packing of the ternary secret and error coefficients are given by their fields and sizes; the security analysis uses only their injectivity and role separation (Assumption F.1). Section D.5 specifies the reference row law, its spread bound, and the canonical signature encoding. The symbols E, M, θ, D_β, and λ denote the quantities of the main construction. Each hash use (key seeds, the matrix, the dither, the message, the challenge, and the reserved public-sample domain) has its own fixed domain prefix, so inputs from different uses never coincide. We call the inputs of one use its domain, for example the key-seed domain or the challenge domain. Each hash use has its own fixed domain prefix, and every variable-length field is preceded by its length, so hash inputs from different uses never coincide and each input parses uniquely. The exact prefixes and input layouts are given in the specification and known-answer test vectors accompanying the reference implementation.[3]

[3] The reference implementation lives at `https://github.com/a16z/pretzel`.

Scope of the specification. The security proof depends on the sampler's complete output distribution, including failures, and on the randomness and reset interfaces stated here. Assumption F.1 requires the implementation to satisfy these conditions.

D.1 Exact parameters

Key generation expands each candidate from a fresh 32-byte key seed, samples its secret and error coefficients from CBD_1, and rejects it unless

$$120{,}000 \le 23{,}716 + 196\|s_0\|_2^2 + 121\|e\|_2^2 \le 244{,}553.$$

The key distribution used below is conditioned on acceptance.

For proposals and transport define

$$E(z) = 9216\|z_0\|_2^2 + 196\|z_s\|_2^2 + 121\|z_e\|_2^2, \qquad M = 65{,}612/65{,}536.$$

The base θ is a dyadic rational with denominator 2^{256} and the following hexadecimal numerator, written most-significant half first:

```
ffffebe71e0d482af9a42878e0fbcb58
f7663bf7f0afda571e6a48849f6572a5
```

Let D_β be the distribution with leading parity vector β and mass proportional to $\theta^{E(z)}$. The ideal proposal is D_0. The implementation approximates these ideal distributions and transport steps using finite cumulative-distribution (CDT) and alias tables, bounded-precision comparisons, and rejection rules; the proof charges the resulting errors separately. For a valid key, $b = E(s)$ lies in $[105{,}500, 230{,}053]$.

D.2 Key generation and dither expansion

1. Draw a 32-byte matrix seed and a 32-byte private PRF key, then expand the public matrix once, before sampling key candidates. For at most 512 trials, draw a fresh 32-byte key seed, expand it with SHAKE256 in the key-seed domain, and sample the 2048 CBD_1 secret and error coefficients. Accept the first candidate that satisfies the key-norm restriction above; if none does, fail. Compute $b = A_0 s_0 + e$ for the accepted candidate. If the entropy source fails, return an error immediately.

2. Draw a 32-byte candidate-start seed. If the request fails, return an error immediately. No public sample stream is read and no row is generated or evaluated.

3. For $i = 0, \ldots, 2047$, add i to the candidate-start seed, read as a little-endian integer modulo 2^{256}, to obtain candidate i. Expand its dither from at most 1536 little-endian 16-bit words, keeping the low 13 bits when they are below 7681, until 1024 values are obtained. If too few values are obtained, advance to the next candidate; any other expansion error is terminal. Return the first candidate whose expansion succeeds. If none succeeds, key generation fails without replacing the core key.

4. Serialize the selected seed with the core key. Public-key preparation expands this same dither.

Dither selection uses only the core public key, the candidate-start seed, and the dither oracle. It is defined for every canonical core, independently of a witness. Its failure outcome is part of the setup experiment. In the ideal independent dither streams, let

$$d_4 = \sum_{j=513}^{1536} \binom{1536}{j} (511/8192)^j (7681/8192)^{1536-j}.$$

This is the failure probability of one bounded expansion. Distinct candidate seeds name distinct streams, so exhaustion has probability d_4^{2048} for a fixed core and independently chosen candidate-start seed. Without any independence between candidates, exhaustion still has probability at most d_4, since it requires the first candidate to fail. The accepted values are independent uniform residues, independently of the accept/reject pattern; conditional on success the selected dither is therefore uniform. With independent key-seed tapes and a functioning entropy source, the key-norm search and dither search give setup failure probability $\eta^{512}+(1-\eta^{512})d_4^{2048}$, where η is the one-candidate key-norm rejection probability. The key-seed replacement error in Section F.3 accounts for using hashed key seeds.

For Pretzel-(5,5), the same selection rule uses the expansion dimensions and cap in Section I.1. Pretzel-(3,3) uses it with the dimensions in Section H.1.

Matrix expansion and key encodings. Use two SHAKE256 output streams, indexed by 0 and 1; the input of each, in the matrix domain, contains the matrix seed ρ_A and the one-byte stream index. Read consecutive little-endian 13-bit fields and retain values below 7681 until each stream has 2048 accepted values. Bits carry over across SHAKE256's 136-byte output-block boundaries: unused bits are neither discarded nor padded. For accepted-value index k within a stream, the polynomial number is $p = 2\lfloor k/256\rfloor + \mathit{stream}$, matrix row $\lfloor p/4\rfloor$, column $p \bmod 4$, and spectral coordinate $k \bmod 256$. The implementation's matrix sampler places no cap on the number of fields it reads. When one stream completes first, the other stream's bit buffer is not reset. The inverse NTT below defines A_0.

Explicitly, let g be the smallest integer in $\{2,\dots,q-1\}$ for which $g^{(q-1)/r} \neq 1 \pmod q$ for $r \in \{2,3,5\}$, and put $\psi = g^{15} \bmod q$. Let $\mathrm{rev}_8(k)$ reverse the eight bits of k and set $\omega_k = \psi^{2\,\mathrm{rev}_8(k)+1}$ for $0 \le k < 256$. The implementation's scalar transform and its inverse are

$$\mathrm{NTT}(a)_k = \sum_{j=0}^{255} a_j \omega_k^j \pmod q, \qquad a_j = 7651 \sum_{k=0}^{255} \mathrm{NTT}(a)_k \omega_k^{-j} \pmod q.$$

Here $7651 = 256^{-1} \pmod{7681}$; the forward transform is unscaled, and both arrays use canonical residues. The ω_k are all the roots of $X^{256}+1$, in the order shown, so polynomial multiplication becomes coordinatewise multiplication in this representation. This fixes the convention independently of any particular optimized transform.

In the ideal model of the matrix streams, the two stream inputs differ in their stream byte and give independent uniform bit streams. Rejecting 13-bit values at least q leaves each accepted value uniform in $\mathbb{Z}_q$, independently of the previous accepted values. The coordinate placement is a permutation of the 4096 accepted values, and the inverse NTT is a bijection on each polynomial. Consequently A_0 is uniform in $R_q^{4\times4}$ in this model; this is not a statement about the concrete seed-expanded matrix.

The untagged core is ρ_A followed by 1024 canonical 13-bit coordinates of $+\mathrm{NTT}(b)$, packed low bits first: $32+1664 = 1696$ bytes. Values at least 7681 are rejected. Append the 32-byte dither seed and prepend the 14-byte parameter-set identifier, given in the specification accompanying the reference implementation, to obtain a 1742-byte public key. The untagged secret body consists of the matrix seed (32), the packed ternary secret and error coefficients (512), the PRF key (32), the raw SHAKE256 hash of the untagged public body (32), and the dither seed (32). Prepending the tag gives a 654-byte secret key. Import rejects forbidden ternary codes, violations of the key-norm restriction, wrong lengths or tags, and any mismatch after recomputing b and the public-body hash. Keys must be generated and decoded under the chosen variant's parameter-set identifier.

D.3 Hash inputs and per-attempt randomness

The dither, message and challenge hashes apply SHAKE256 to inputs that start with a common prefix, the length-prefixed parameter-set identifier and core public key, followed by the domain prefix of that use. The dither input then contains the dither seed; the message input contains the dither seed, the context string and the message; the challenge input contains the dither seed, μ, the salt ρ and U. For these three inputs the domain prefix sits directly after the common prefix, and no domain prefix is an initial segment of another. All variable-length fields are preceded by their lengths, so each input is parsed uniquely. Each key-seed input has 53 bytes. Rank-four setup reads no public-row input. Every honest hash input has at most 4096 bytes, apart from the context and message fields of the message-hash input. SHAKE256 gives 64 bytes for μ and 32 bytes for the challenge digest. In the challenge input, U is represented by 1024 one-byte labels; its compact local-index encoding is used only in the signature. The public-body integrity hash is raw SHAKE256 of that body, which has 1728 bytes, while every challenge input has 2916 bytes. Matrix expansion uses exactly the input specified above, which contains no parameter-set identifier.

To fold the digest, take its first 255 bits in little-endian order, append a zero, and complement all 256 bits if the weight exceeds 128. The resulting family is

$$C_{255} = \{c \in \{0,1\}^{256} : |c| < 128\} \cup \{c \in \{0,1\}^{256} : |c| = 128,\ c_{255} = 0\}.$$

For attempt $j \in \{0, \ldots, 511\}$, request 32 bytes of fresh entropy. The attempt encodes six domain-separated streams, for the salt, lead, secret, error, transport, and dummy roles. The signer reads no output from the dummy stream: the transport stream supplies one coin for each of the 128 positions, and each position not used by the challenge spends its coin on a dummy test. Each stream's hash input encodes the parameter-set identifier, the schedule-domain identifier, the PRF key, the untagged public body, μ, the attempt index j, and the entropy. The schedule-domain identifier names this sampling schedule; it is given in the specification accompanying the reference implementation. The salt is the first 32 bytes of the salt stream. The input contains no call identifier, so repeating the entire encoded input repeats the attempt's random streams.

Sampling the mask. The measured Pretzel-(4,4) implementation arranges its sampling tables by byte position. It reads fixed memory locations and selects entries inside processor registers. Most coefficients are determined from a few random bits. Unfinished choices are saved, up to a fixed limit, and completed using further bits. Exceeding that limit marks the attempt for rejection. The parts of the mask corresponding to the secret and error are sampled from fixed allocations of random bits.

Each attempt prepares a fixed amount of randomness from the private SHAKE streams, but sampling the first polynomial can require more.

D.4 Signing and verification

Compute μ once per call. In each attempt, sample the finite proposal y, form $t = Hy + \delta$, set $U = Q(t)$, and hash and fold the challenge. Starting at $z = y$, process the nonzero challenge positions using $v = X^j s$, $a = \langle z, v \rangle_W$, and $b = E(s)$. Use the probabilities $f(a,b)$ and $g(a,b)$ of Section 4.3.2 to select $z - v$, $z + v$, or abort. Decisions follow the bounded-precision rule of Section 4.3. The attempt is rejected if a deferred check does not confirm its step, including when the longer polynomial leaves the case unresolved; if a recomputed integer logarithm remains ambiguous; or if more than eight of the 128 integer logarithms need recomputation or more than six steps are deferred. Pad to 128 steps with the $128 - w$ dummy tests of Section 4.3, where w is the challenge weight; a dummy test passes when its position's coin lies below a 384-bit lower bound on λ from (23). The ideal transport uses independent coins at each step. The implementation instead gives each of the 128 positions its own independent high 128 bits but a common low 256-bit tail. The finite-to-ideal comparison of Appendix F charges the resulting correlation.

Finite decisions for Pretzel-(4,4). The implementation first prepares the coins for all 128 positions. For a nonzero 384-bit coin representing $u \in (0, 1)$, a mixed-precision logarithm evaluator locates $\log_\theta(Mu)$ between consecutive integers. Away from an integer boundary it returns $\ell = \lceil \log_\theta(Mu) \rceil$; close cases are refined by bounded power comparisons. The all-zero coin uses the convention $\ell = 0$. At most eight logarithms can be saved for recomputation. Exceeding this limit or a comparison that remains unresolved marks the attempt for rejection.

At a transport position let $a = \langle z, X^j s \rangle_W$ and $b = E(s)$, with z denoting the response before that position's update. As in SHUTTLE [33, Section 3.3], the integer comparison tests membership in

$$P = \left[\ell \in \bigcup_{r \geq 0} \left(b(2r+1)^2 + 2(2r+1)|a|,\ b(2r+2)^2 + 2(2r+2)|a| \right] \right].$$

Six search comparisons suffice on the certified input domain. A separate short polynomial evaluates the correction comparison. An active position is marked when this comparison lies in its uncertainty band, even if P already holds. A marked position provisionally chooses plus if P and minus otherwise; these signs are swapped when $a < 0$. For at most six marked steps, save a, ℓ, and all six original coin words. After processing all 128 positions, each saved step is checked with the longer polynomial on those saved values. If F denotes acceptance of that finite fine comparison, the step rejects exactly when $\neg P \wedge \neg F$. More than six marked steps also cause rejection.

For each active position, the initial inner product is carried through the position-routing network. Its current value is recovered by adding the precomputed Gram contributions $\langle s, X^k s \rangle_W$ for previously recorded signed updates, so it equals the pre-update inner product above under the integer and routing contracts. The updates to the response itself are materialized after the position loop. Inactive positions perform dummy tests on the coins assigned to those positions. Failures accumulate in an abort flag; every position is processed and every final check runs.

Under the coarse-to-fine error bound, deferred decisions that survive all checks agree with immediate finite fine decisions on the same coins. Logarithm-preparation failures and attempts exceeding either limit on saved values are charged separately. This statement concerns the returned signature or abort and the prescribed state after reset. The sampling error bound, including the shared coin tail, remains subject to Assumption F.1.

The finite sampler of the measured Pretzel-(4,4) signer is fixed by the following choices. The fine comparison uses a degree-34 correction polynomial with coefficients rounded to multiples of 2^{-192}; the coarse comparison re-expands it to degree 6 with coefficients rounded to multiples of 2^{-62}, and a step is deferred when the coarse value lies within $2^{-34} + 2^{-57}$ of the threshold. A dummy test accepts when its 384-bit coin lies below a 384-bit lower bound on $\lambda = Z_{\text{odd}}/(MZ_{\text{even}})$. The mask sampler saves at most 6 unfinished choices for the first polynomial, and at most 7 blocks of eight coefficients, plus one further block at a second level, for the secret and error parts. The exact tables and coefficients are generated by, and published with, the reference implementation.

Transport preserves t modulo q. Form $x = t - z_e = A_h z_h + \delta$, reconstruct r_e, and apply every public cap. Before release, the signer also checks that the response quotients and residual exceptions fit in their encoding fields. Encode only after all checks pass; if the encoder nevertheless fails, reject the attempt. Each retry requests new entropy, up to 512 times. If all attempts reject, signing fails. A failure of the entropy source returns an error immediately and does not count as an ordinary rejection. In the security experiment, private intermediate states stay hidden, and the reset conditions of Appendix E apply. The measured implementation does not verify signatures before releasing them.

The signature is $\rho_{32} \parallel head_{1160} \parallel local_{169}$. Section 4.5 specifies the Rice quotients and remainders, the local-label exception positions, and the canonical padding. The challenge hash consumes the 1024 recovered labels, one byte each.

Verification starting from a stored public key first validates and unpacks it. It then requires the exact signature length and canonical fields, computes the challenge with its encoded hash input and the public relation, reconstructs r_e, and checks all caps. Finally, it checks that the leading parity vector belongs to C_{255} and equals the folded digest. Verification with unpacked and stored public keys use this same acceptance predicate.

D.5 Encoding and Public-Filter Certificates

Section 4.5 specifies the Rice-coded encoding of z_h and the Elias–Fano local-label field. The sum of the head quotients is at most $K_h = 1088$, and the local-label field permits 50 exceptions, with four low bits per position. The coefficient caps are $(288, 255, 450)$ on (z_0, z_s, r_e); the residual cap 450 ensures that the four-candidate list contains every label satisfying the residual cap. The challenge hash still takes the recovered labels as input, one byte each. The rejection bound below holds uniformly over every fixed public key, dither, and supplied challenge.

For fixed public key, dither, salt and commitment, Lemma 4.6 gives the canonical bijection without a hash comparison, also after restriction to a supplied challenge; Remark B.2 explains why decoding can be run reversibly.

Lemma D.1 (Complete local recovery lists)**.** For $q = 7681$ and the 34-center quantizer of Section 4.2, put $j = Q_{q,34}(x)$, and choose $s \in \{-1, 1\}$ toward the nearer adjacent center, with a tie toward -1. For every $x \in \mathbb{Z}_q$, the four distinct labels $(j, j+s, j-s, j+2s) \bmod 34$ contain every u satisfying $|\operatorname{ctr}_q(C(u)-x)| \le 450$. For the 64 Pretzel-(5,5) centers $c_u = 120u + 60 + \mathbf{1}_{\{u\neq 0,\ u\equiv 0 \pmod 2\}}$, the set of labels satisfying that cap has seven or eight elements for every x. Ordering it by $(|\operatorname{ctr}_q(c_u - x)|, \operatorname{ctr}_q(c_u - x))$ is unambiguous, puts the negative residual first in a tie, and gives the public Pretzel-(5,5) recovery row. An absent eighth entry rejects.

Proof. The following finite integer certificate uses every context, including the modular boundary. Set $\operatorname{ctr}_q(v) = ((v+3840) \bmod 7681) - 3840$. Generate the 34 centers from the cell endpoints $a_j = \lceil 7681j/34 \rceil$, alternating lower and upper midpoints only for even-sized cells, beginning with the lower midpoint. For each integer $x = 0, \ldots, 7680$, form the displayed four-label row using the two centered adjacent distances. Compare all 34 centers with the residual cap and record any label satisfying the residual cap outside that row. The certificate has 261,154 comparisons, no omitted pair satisfying the cap and no row with a duplicate label. In fact the smallest absolute residual of any excluded label is 451.

For Pretzel-(5,5), at each of the same contexts enumerate the 64 displayed centers, retain precisely the labels satisfying the cap and sort by the displayed pair. The 491,584 comparisons give only row lengths seven and eight. The centers are distinct modulo q, so neither equal residuals nor duplicate labels occur. Equivalently, merge the centers immediately to the left and right of x, repeatedly choosing the smaller absolute residual and choosing the left center in a tie. Eight selections suffice. Store a selected label only when its residual passes the cap; otherwise store the reject sentinel. Exhaustive comparison of all 61,448 positions gives exactly the sorted rows, including the sentinel. These integer loops, the complete row witness and their outputs form the accompanying local-label certificate. No probability or successful-signing conditioning is used in either enumeration. □

D.5.1 Rounding error for a uniform input

Use the interval centers specified in Section 4.2. For $(q, L) = (7681, 34)$ there are three intervals of size 225 and 31 of size 226. The alternating choice reduces the midpoint bias but does not make the error law exactly symmetric: for uniform $x \in \mathbb{Z}_q$, represented canonically, $\mathbb{E}[C(Q_{q,L}(x)) - x] = -113/7681$. If

$E_Q = C(Q_{q,L}(x)) - x$ for uniform canonical x, then

$$\Pr[E_Q = e] = \frac{1}{7681}\begin{cases} 34, & -112 \le e \le 112, \\ 16, & e = -113, \\ 15, & e = 113, \\ 0, & \text{otherwise.} \end{cases}$$

Every interval contributes once to each error from -112 through 112. The lower and upper midpoints of the even intervals contribute the two endpoints. Hence $\mathbb{E}[E_Q^2] = 32{,}668{,}639/7681$.

D.5.2 Choice of L and the four-label bound

This subsection supports the parameter choice of Section 4.2, with $q = 7681$ and residual bound $B = 450$. A label j is possible for an estimate x_i when it satisfies (11).

Interval sizes and centers. For $L = 34$, each interval contains 225 or 226 integers. The first contains the 226 integers from 0 through 225, and the second runs from 226 through $226 + 226 - 1 = 451$. Its midpoint is 338.5, rounded upward to $C(1) = 339$ by the alternating rule. The only intervals of size 225 have labels $11, 22, 33$.

Nearby choices of L. The table compares nearby choices of L, keeping q and B fixed.

Intervals	Max rounding error	Max possible labels
33	116	4
34	113	4
35	110	5

Compared with 33 intervals, 34 reduce rounding error while retaining four possible labels. A smaller rounding error leaves more of the coefficient bound for z_e itself, which affects how often signing retries: by (9), every coordinate with $|z_{e,i}| \le B - 113 = 337$ passes the check $|r_{e,i}| \le B$ for $L = 34$, against $|z_{e,i}| \le 334$ for $L = 33$. Moving to 35 would require distinguishing five labels in some cases, which no longer fits in a two-bit local index and would change the encoding of Section 4.5.

A worked example. The table follows the calculation for an estimate $x_i = 1000$.

Label j	Coefficient values	Center $C(j)$	$C(j) - 1000$
0	0–225	112	−888
1	226–451	339	−661
2	452–677	564	−436
3	678–903	791	−209
4	904–1129	1016	16
5	1130–1355	1243	243
6	1356–1581	1468	468

Only labels 2, 3, 4, 5 give a reconstructed coefficient between -450 and 450. All other centers are too far from 1000, including those reached by wrapping around modulo q. The signature therefore needs to distinguish only these four labels.

Lemma D.2 (Four-label bound)**.** For $(q, L, B) = (7681, 34, 450)$ and every $x_i \in \{0, \dots, q-1\}$, at most four labels j satisfy (11). The bound is sharp in B: with bound 451, five labels are possible at $x_i = 2598$.

Proof. Extend the centers periodically by $C(j+34) = C(j)+q$, so that distances modulo q become ordinary distances between integers. Since the only intervals of size 225 have labels $11, 22, 33$, any five consecutive intervals contain at most one of that size. The first and last midpoints of five consecutive intervals are therefore at least $113 + 225 + 226 + 226 + 113 = 903$ apart. Rounding both midpoints to integers can reduce this separation by at most one, so five consecutive centers span at least 902. Five labels satisfying (11) would need five centers in the range from $x_i - 450$ to $x_i + 450$, of width 900, which is impossible.

For sharpness, at $x_i = 2598$ the centers for labels 9 through 13 give differences $-451, -226, 0, 226, 451$, so a bound of 451 admits five labels. □

Lemma D.1 checks exhaustively that the encoding's four-entry recovery list contains every label satisfying (11).

D.5.3 Rejection caused by encoding limits

Let D_c denote Pretzel's ideal product response distribution for a prescribed leading parity vector, before any caps are applied. For this encoding, a certificate computed with directed rounding bounds the additional rejected mass by 0.000220, uniformly over every fixed key, dither, and c. The certificate adds the following bounds for coefficient-cap failure, quotient overflow and residual-exception overflow, respectively:

$$\begin{aligned} \Pr[\text{coefficient cap fails}] &< 41 \cdot 10^{-6}, \\ \Pr[\text{head caps pass, } \textstyle\sum_i a_i > K_h] &< 38 \cdot 10^{-6} \quad (t = 23/20), \\ \Pr[\text{residual caps pass, more than 50 exceptions}] &< 83 \cdot 10^{-6}. \end{aligned}$$

Their sum is below the stated rejection bound. For the quotient event it uses the capped moment

$$M_{w,B,k,e}(t) = \frac{\sum_{|z| \le B,\ z \equiv e\ (2)} \theta^{wz^2} t^{\lfloor v(z)/2^k \rfloor}}{\sum_{z \equiv e\ (2)} \theta^{wz^2}}, \qquad t > 1.$$

The denominator is the full normalizer. The certificate replaces it by a lower bound computed from a truncated sum, which can only enlarge the bound on the moment. Independence of the coefficients and integrality of the quotient sum give

$$\Pr[\text{head caps pass, } \textstyle\sum_i a_i > K_h] \le t^{-(K_h+1)} \max(M_0, M_1)^{256} M_s^{1024}.$$

Here M_0 and M_1 use weight 9216, cap 288, and three remainder bits, with the indicated parity. The moment M_s uses weight 196, cap 255, and six remainder bits, without a parity restriction. Conditioning on the head fixes every residual context. The omitted coefficients are independent, and each gives an exception with probability at most $7/250$, so the number of residual exceptions is dominated by a binomial count. The argument does not need independence after the public filters are applied.

D.5.4 Reference rows and rank-four setup

Rank-four setup draws no test rows. The following reference experiment specifies the row-quality statement used for availability. Fix a core (A_0, b), dither δ, and leading parity vector β. Draw 2304 independent uniform 32-bit words: 256 for the leading coefficients, 1024 for the other head coefficients and 1024 for the errors. For each parity or weight, allocate $\lfloor 2^{32} p_z \rfloor$ words to integer z, where p_z is its full ideal Gaussian probability under θ from Appendix D. Cumulative endpoints are exclusive; a word beyond the

last endpoint fails without resampling. The EVEN, ODD, SECRET and ERROR table counts equal these individual floors exactly, with zero entries omitted. Write the resulting head as $h = (h_0, h_1, h_2, h_3, h_4)$ and set $x = -bh_0 + A_0(h_1, h_2, h_3, h_4)^T + \delta$. For each coefficient set $U_i = Q_{q,34}(x_i + e_i)$ and $r_i = \mathrm{ctr}_q(C(U_i) - x_i)$. Let F mean that all samples exist, the head caps $(288, 255)$ and residual cap 450 hold, every label is recoverable, and

$$\|h\|_2^2 \le 2537649, \quad \|r\|_2^2 \le 9412624, \quad \sum_k W_{\mathrm{HEAD}}(|h_k|) + \sum_i W_{\mathrm{RES}}(|r_i|) \le 30850912.$$

The score tables are the integer secant tables of Appendix B.2. Also require head Rice quotient sum at most 1088, with shifts 3 and 6 on the leading and other head coefficients, and at most 50 residual local indices at least 2. Thus F includes the complete rank-four public response and encoding predicate. This mathematical row law is used through its ideal-product consequence; rank-four key generation draws no such rows.

Lemma D.3 (Rank-four spread and row quality)**.** Let $q = 7681$ and $A_0 \in (\mathbb{F}_q[X]/(X^{256}+1))^{4\times 4}$. For output coordinate $i = (r, a)$ and input coordinate $k = (c, b')$, define

$$a_{i,k} = (-1)^{[b'>a]}(A_0)_{r,c,(a-b') \bmod 256}.$$

For a residue v put $|v|_q = \min(v \bmod q, (-v) \bmod q)$. The condition $G(A_0)$ is

$$\#\{k : |ta_{i,k} + sa_{j,k}|_q = u\} \le 48 \quad (i \ne j,\ (t,s) \in \mathbb{F}_q^2 \setminus \{(0,0)\},\ 0 \le u \le 3840).$$

For every A_0 satisfying G, every $b \in R_q^4$, every fixed $\delta \in R_q^4$ and every fixed $\beta \in \{0,1\}^{256}$, the reference row above satisfies

$$\Pr[\neg F \mid A_0, b, \delta, \beta] \le \varepsilon_4 := 0.292944128348880 < 0.293.$$

The probability is over fresh independent row words. The dither and parity are fixed before these words; this is a pointwise statement, not a statement about choosing the dither after observing a row. The ideal product law D_β has at least the same successful mass. For 4096 independent uniform matrix coefficients,

$$\Pr[\neg G] \le \varepsilon_G := \frac{(q+1)^2}{2}\left(1024\binom{512}{25}(2/q)^{25} + 1536\binom{1024}{49}(2/q)^{49}\right) < 1.423 \cdot 10^{-37}.$$

This exceptional probability concerns an honestly expanded matrix in the ideal uniform-matrix model of Appendix D.2. The vector b may depend on A_0, including $b = A_0 s_0 + e$.

Proof. First condition on all coordinate samples existing. This event factors by coordinate, so the normalized coordinates are independent. The normalized SECRET law μ is symmetric. Put $\phi(v) = \sum_z \mu(z)\cos(2\pi vz/q)$ and sort upper bounds on its 3841 absolute-frequency magnitudes as $B_1 \ge \cdots \ge B_{3841}$, including frequency zero. Each bound is at most one. The accompanying exact certificate gives

$$\left(\prod_{j=1}^{21} B_j^{48}\right) B_{22}^{15} \le \nu_4 := 3087/2^{160}.$$

Fix the whole leading block h_0. For a nonzero pair frequency (t, s), independence makes the magnitude of the Fourier coefficient of (x_i, x_j) the product of the SECRET factors $|\phi(ta_{i,k} + sa_{j,k})|$. The fixed term $-bh_0 + \delta$ changes only its phase. Condition G permits at most 48 factors in any absolute-frequency bin. Among 1023 factors the largest possible product therefore takes 48 copies of each of the first 21 upper bounds and 15 of the next. This still applies after fixing any one SECRET coordinate: it contributes only a phase and removes one factor. For all 1024 coordinates, deleting any factor of magnitude at most one can only increase the product. Consequently every nontrivial Fourier coefficient, with or without this conditioning, has magnitude at most ν_4.

Fourier inversion bounds each point's deviation from uniform by $(q^2-1)\nu_4/q^2$. Summing absolute deviations and halving gives

$$d_{\mathrm{TV}}((x_i,x_j),U_q^{\otimes 2}) \le \tau_4 := (q^2-1)\nu_4/2 < 6.231\cdot 10^{-38}.$$

Marginalization gives the single-coordinate bound. Mixing over the leading block preserves it, including conditional on one leading coordinate. The deletion argument gives it conditional on any one other head coordinate. No independence of all residual coordinates is asserted.

For moment calculations, replace a statistic by zero outside its coordinate cap. This does not change it when all caps pass. Charge missing samples and cap failures separately. With an independent uniform context and normalized ERROR draw, let μ_Y, v_Y be the mean and variance of a capped residual statistic $Y \in [0,B]$. Adding independent errors and applying a deterministic function preserve the total-variation bounds. Thus each first moment differs from its reference value by at most $B\tau_4$, and each second or pair-product moment by at most $B^2\tau_4$. Products of marginal means differ by at most $2B^2\tau_4$, because both means lie in $[0,B]$. It follows that

$$M_Y := 1024(\mu_Y + B\tau_4) \ge \mathbb{E}\sum_i Y_i, \qquad V_Y := 1024 v_Y + 3\cdot 1024^2 B^2\tau_4 \ge \mathrm{Var}\sum_i Y_i.$$

For a capped head score $H_k \in [0,H_{\max}]$, conditioning on that coordinate gives $|\mathbb{E}[Y_i \mid h_k] - \mu_Y| \le B\tau_4$. Expanding the covariance around μ_Y therefore bounds it by $2H_{\max}B\tau_4$ in absolute value. The head coordinates are independent, so for the combined score add their means and variances to M_Y, V_Y and add $4\cdot 1280\cdot 1024 H_{\max}B\tau_4$ to the variance. For each leading mean and variance take the larger of the EVEN and ODD values; this covers every prescribed parity pattern.

All reference moments are finite sums: for each $x \in \mathbb{F}_q$ and error table atom e with word count n_e, evaluate $r = \mathrm{ctr}_q(C(Q(x+e)) - x)$ and its local index, weight by $n_e/(q\sum_e n_e)$, and accumulate the first and second moments. The three statistics are capped r^2, capped $W_{\mathrm{RES}}(|r|)$, and the capped indicator of local index at least 2, with maxima 450^2, $\max_{a\le 450} W_{\mathrm{RES}}(a)$ and 1. The exact moments give the following bounds on their sums; the score includes its head terms:

	M upper bound	V upper bound	$T+1$
residual norm	7887638.684	99286496761	9412625
combined score	29331180.304	754072431650	30850913
exceptions	5.541597	5.511608	51

The certificate uses exact rational moments, not the displayed rounded summaries. For $\mathbb{E}Z \le M < T+1$ and $\mathrm{Var}\,Z \le V$, Cantelli's inequality gives $\Pr[Z \ge T+1] \le V/(V+(T+1-M)^2)$; this bound is monotone in M and V.

For head norm and head Rice quotients, use the original subprobability word masses and intersect coordinate cap acceptance. For a rational base $b_0 > 1$, independence and Markov give

$$b_0^{-(T+1)}\max(m_{\mathrm{EVEN}}, m_{\mathrm{ODD}})^{256} m_{\mathrm{SECRET}}^{1024}, \quad m_L = 2^{-32}\sum_{|z|\le B_L} n_L(z) b_0^{f_L(z)}.$$

For norm use $f_L(z) = z^2$, $T = 2537649$ and $b_0 = 10000320/10000000$. For quotients use the signed Rice integer $v(z) = 2z$ if $z \ge 0$, and $v(z) = -2z-1$ otherwise; $f_L(z) = \lfloor v(z)/2^{k_L}\rfloor$, $k_L = 3$ or 6, $T = 1088$ and $b_0 = 11500/10000$.

The missing-word union bound is $161/1048576$. There are no leading cap-tail words, 100 SECRET cap-tail words per coordinate, and at most 12 residual cap-tail words per context; hence the cap union bound is $7/262144$. Enumeration of every context and all 34 labels gives no unrecoverable label within residual cap

450, including labels of zero sampling mass. Use the unconditional cap and missing bounds, and bound the other events conditional on samples existing without conditioning on caps. Multiplying those conditional probabilities by a number at most one only reduces them. The six upper bounds, rounded upward, are

missing samples	0.000153541564942
coordinate caps and recovery	0.000026702880860
head norm	0.002986304974391
residual norm	0.040945084249347
combined score	0.246134510974904
both encoding capacities	0.002697983704435

The encoding term adds the head quotient bound to the exception Cantelli bound, without a binomial assumption for dependent residuals. The displayed upper bounds sum to $0.292944128348879 < \varepsilon_4$. The finite atoms are downward-rounded ideal Gaussian masses; multiplying and summing over F proves the ideal-product assertion without normalizing finite successful mass.

An accompanying checker and certificate verify these numerical steps. The Fourier factors are enclosed with outward-rounded arithmetic, and sorting their upper endpoints and rounding the product upward gives ν_4; the head moments use outward 512-bit arithmetic, and all other moment and tail calculations are exact rational. Every numerical step is therefore a finite enclosure.

We now prove the matrix probability statement. Scaling a nonzero pair frequency permutes absolute bins, so only $q+1$ projective directions are needed. Simultaneously shifting two output coefficients negacyclically permutes the input columns and changes their individual signs, with a possible whole-output sign change. Individual signs leave absolute projections unchanged; a whole-output sign permutes directions. Exchanging the outputs also permutes directions. The coefficient formula therefore reduces all pairs to $(r, 0), (r, d)$ for $1 \le d \le 128$ (512 representatives), and $(r, 0), (s, d)$ for $r < s$, $0 \le d < 256$ (1536 representatives).

For a cross-polynomial representative, the 1024 projections in a fixed nonzero direction are independent uniform residues, since they use distinct coefficients in each of two independent matrix rows. An absolute bin has probability at most $2/q$, including the zero bin. A union over 49-element subsets gives $\binom{1024}{49}(2/q)^{49}$. For a same-polynomial representative with displacement d, coefficient pairs are signed edges of cycles of length $256/\gcd(256, d)$, which is even. Alternating edges give two matchings; for $d = 128$ these are the two parallel edges between each pair of vertices. Across four input polynomials each matching has 512 projections of disjoint coefficient pairs. Every nonzero direction gives independent uniform projections within a matching, even when one direction coordinate is zero. A bin with at least 49 points has at least 25 in one matching, so its probability is at most $2\binom{512}{25}(2/q)^{25}$. No independence between matchings is used. Union over representatives, $q+1$ directions and $(q+1)/2$ absolute bins gives precisely ε_G. The condition depends only on A_0, so the exceptional event already covers every fixed b, dither and parity without another union bound. □

The order of quantifiers is necessary. After observing a head, choosing $\delta = -A_h h$ makes all contexts zero. Every quantizer center then has centered magnitude at least 112, so a complete row has residual squared norm at least $1024 \cdot 112^2 > 9412624$ and fails. This does not contradict the bound for each dither fixed before the row words.

Exhausting the signing attempts. Setup can fail through key-norm or bounded-expansion exhaustion (Appendix D.2). For a matrix satisfying G of Lemma D.3, an ideal public-sampler attempt with a fresh independent challenge and row has successful mass greater than $p_4 = \lambda^{128}(1 - 0.293)$: the transport gate has mass λ^{128} and the complete public filter passes with probability above $1 - 0.293$ for each challenge. For m fresh attempts, induction on conditional failure probabilities bounds all-failure probability by $(1 - p_4)^m$.

This statement does not treat exact-name replays as fresh attempts. In the ideal uniform-matrix model, the probability that any of K predrawn honest matrix expansions violates G is at most $K\varepsilon_G$, by a union bound, including seed collisions. The comparison of real and simulated signing in (F.1) changes the probability of an event on the complete output by at most Δ_F. Consequently, under the assumptions of Theorem 5.6 and the ideal uniform-matrix model, for at most S signing requests on honestly generated keys, let X be the event that some request exhausts all 512 attempts while all attempted private-stream names are distinct. Then

$$\Pr_{\text{real}}[X] \le S(1-p_4)^{512} + K\varepsilon_G + \Delta_F.$$

The first term is the union bound for exhausted requests in the exact public-sampler experiment when every used matrix satisfies G; $K\varepsilon_G$ bounds the probability that any of the K predrawn matrices violates G. The last term bounds the change of this complete-output event between that experiment and the real one. The row bound holds for every public column and fixed dither, so it also covers the unrestricted core at the public endpoint. Setup failures and entropy errors are distinct from exhausting the attempt cap; the inequality does not condition on their absence. This separates row quality from finite-sampler error and from challenge programming. Theorem 4.4 concerns every returned signature, including for a matrix outside G.

D.5.5 Public checks that responses fit in the encoding

Proposition D.4 (Public-filter transfer)**.** Fix the public data and prescribed challenge. Suppose the implementation's complete output distribution for one attempt is within total variation ε of Pretzel's ideal response distribution, both including aborts. Applying the deterministic public check that the response fits in the chosen compact encoding and canonical encoder to both distributions preserves the bound ε. In particular, rejecting a response that does not fit adds no statistical-distance term. The same holds for the joint distribution of the response and all other retained data whenever the comparison supplies those data before this public map.

Proof. The coefficient tests and checks that the response fits in the encoding are functions of the reconstructed response and public data. Map a failed test to the common abort symbol and a response that passes to its unique encoding. Total variation cannot increase under a deterministic map. The argument works with subprobability masses and their complementary abort mass, without conditioning on success. For joint distributions, apply the same map while keeping all other data. At each supplied challenge, the chosen parameter set's numerical response domain is a subset of Pretzel's domain of reconstructed responses, so its extraction inequalities and upper bound on secant-feature norms remain valid. □

E Comparing Real and Simulated Signing, Including Aborts

All distances are trace distances (half the trace norm) between the views of the permitted observer, extended to interactive observers whose future queries are counted. The observer has no access to purifications of hidden oracle tables, to the private random tapes of failed attempts, or to oracle values that reprogramming has replaced.

Theorem 4.3 shows that the ideal sampler's accepted responses have the subprobability distribution αD_c, where $\alpha = \lambda^{128}$ and λ is the ratio defined in (23). We use the parameters of Section 4.1, including a weighted squared norm $E(s)$ at most 230,053 and envelope 65,612/65,536. The term D_{core} charges the finite implementation's deviation from this distribution. Assumption F.1 states the conditions on the finite sampler and random streams of the 1361-byte implementation.

E.1 All signing outputs and the retained quantum state

Fix the classical public context and an admissible witness s with $Hs = 0$. Recall that an attempt's *name* is its complete encoded input to the private randomness streams. At the first occurrence of each attempt name, we compare the two experiments before either reads that attempt's private randomness. At this point the proposal and transport coins must be fresh and independent of all retained quantum state; we call this retained state the *entry reference*. An attempt whose name repeats an earlier one replays the cached result. We introduce this replay representation before any selective programming, while the original public oracle is still fixed. For each fixed oracle table, immutable key, and complete attempt name, the logical reset described below makes recomputation deterministic with the same full outcome. Replaying that outcome is therefore exact pointwise, including on rejection. Calls to the entropy source and updates to the retry count are preserved. Later hybrids keep this replay rule rather than recomputing failed attempts against a modified challenge table.

Write δ for the fixed dither, so the commitment before quantization is $t = Hy + \delta$. All accepted updates add or subtract rotations of s, so $Hz = Hy$ and

$$U = Q(t) = Q(Hz + \delta).$$

In particular, the commitment that the implementation retains is the one computed from the final response, not an independently resampled one. After the selective-answer comparison below, the raw challenge R and the salt ρ are independent uniform coins, and transport uses R only through $c = \mathrm{Fold}(R)$. Thus, for every test function f on complete successful outputs, the integral of f against the submeasure of successful outputs is

$$\alpha\, \mathbb{E}_{R,\rho,z\sim D_{\mathrm{Fold}(R)}} [\mathbf{1}_{F_v(R,\rho,z)} f(R, \rho, X(\mathrm{pk}, \mu, \rho, U(z)), \sigma_v(R, \rho, z))].$$

The event F_v incorporates the final public response caps and the canonical-encoding failures, all deterministic in the displayed variables. Tails of the proposal sampler, intermediate arithmetic overflow, and exhaustion of the finite sampler are accounted for in the finite-to-ideal coupling of complete outputs, including aborts, not in this filter. The raw challenge, the encoded challenge input, the salt, the signature bytes, and the controlled reprogramming of the current oracle are all deterministic functions of these variables. Hence the equality holds for their joint classical outputs and retained quantum state, not just for the distribution of signature bytes.

On failure, the *logical reset* requirement leaves the previously existing active state unchanged, apart from a marker recording the rejection and the attempt name; no later operation reads the private fields of the failed attempt. The failure branch is thus the same channel on the entry state in both experiments, multiplied by the complementary mass, without normalization. Tensoring the classical entry blocks with arbitrary reference states extends this equality to completely positive (CP) maps with classical outputs. Repeats of an exact name use the public cached result or the rejection marker. Induction over first occurrences and repeats shows that the two interactions remain equal throughout, under the stated restrictions on access to failed-attempt data.

E.2 Changing an oracle at hidden inputs

Let O_w and O be name-preserving oracle gates, with $D_w = O_w - O = P_w D_w P_w$, $\|D_w\| \le 2$, and $\mathbb{E}_w P_w \le pI$. The common initial state and the circuit between oracle calls are independent of the protected label w; coins of the table and tapes that do not depend on the label may be fixed first. The label is averaged after its protected control is traced out, and it is never given to the observer. For Q calls, the final distance seen by the observer, averaged over w, is at most $\min(1, 4Q^2 p)$.

The following mean-displacement argument gives this quadratic bound rather than a square-root bound. Purify the common circuit. Let ψ_t be its baseline state and ζ_t the displacement of the modified state from it.

In the telescoping expansion of ζ_t, each summand is a modified unitary suffix applied to $D_w\psi_j$, where the baseline query state ψ_j is independent of w. Consequently

$$\mathbb{E}_w\|\zeta_t\|^2 \le t\sum_{j<t}\mathbb{E}_w\|D_w\psi_j\|^2 \le 4t\sum_{j<t}\langle\psi_j,\mathbb{E}_w P_w\psi_j\rangle \le 4t^2p.$$

The modified suffix may depend on w; only the fact that it preserves norms is used. The two-sided support condition gives $\|\mathbb{E}D_w\| \le 2p$ by applying Cauchy–Schwarz to $\mathbb{E}\langle P_w u, D_w P_w v\rangle$ for unit vectors u, v. The same argument with v replaced by the label-dependent displacement gives, by duality, $\|\mathbb{E}D_w\zeta_t\| \le 2\sqrt{p\mathbb{E}\|\zeta_t\|^2} \le 4tp$. The recursion

$$\zeta_{t+1} = V_t O\zeta_t + V_t D_w\psi_t + V_t D_w\zeta_t$$

therefore yields $\|\mathbb{E}\zeta_Q\| \le 2Q^2p$. Expanding the averaged density difference gives half trace norm at most $\|\mathbb{E}\zeta_Q\| + \mathbb{E}\|\zeta_Q\|^2/2 \le 4Q^2p$. Measurements, aborts, and controlled inactive calls are dilations common to both circuits. The proof fails if the active circuit keeps caches computed from the numerical values of hidden labels, which is why the normalization requirement stated later in this appendix is essential.

We apply this lemma to two families of streams. First, the private signing streams of each key are encoded with that key's 256-bit PRF key, which plays the role of the label w in the lemma above. Predraw K independent PRF keys, one for each key-generation call the adversary may make, and split once on whether they are distinct. Conditional on distinctness they are sampled without replacement, independently of the baseline circuit, in which intended reads bypass the numerical PRF keys and use streams indexed by key-generation call instead (this normalization is made explicit later in this appendix). A parsed query input names at most one label, so $p = K/2^{256}$; the disjoint private roles, the implementation's six and the private stream used only by the public sampler in the proof, do not multiply this. Replace the complete bounded streams for all values of the remaining input fields, not only for keys that are eventually accepted. Swapping these coordinates with independent persistent tapes preserves the uniform distribution of the oracle table; collisions follow the convention for repeated PRF keys, under which a label drawn by several key-generation calls keeps the streams of the first such call, and are charged once, by δ_G. The private stream used only by the public sampler in the proof belongs to this family indexed by PRF keys, so this replacement already covers its rows: afterwards the sampler reads streams indexed by key-generation call that do not depend on the numerical PRF keys. Its reads need no separate replacement. The private-family and key-seed replacements are made one after another, not as a simultaneous correlated installation.

Second, for the key-seed domain (the hash inputs that carry the key-seed domain prefix and expand a 32-byte key seed into the random stream of one key candidate; Appendix D), predraw $B_* = 512K$ seed labels, keep the branch on which they collide, which has probability δ_K, and use $p = B_*/2^{256}$ on the branch on which they are distinct. Each intended CBD sampler reads its own independently cached 512-byte tape; the coordinate-swap argument covers the full stream, including tails read by the adversary. The exact bit-pair CBD map produces the prescribed independent trits. The numerical seeds appear only in protected naming registers. The same lemma proves $e_{KD} = \delta_K + (1-\delta_K)\min(1, 4B_*Q^2/2^{256})$. A public core key produced by the bounded setup and its trits (the secret and error coefficients in $\{-1, 0, 1\}$ that the CBD map produces from these tapes) may depend on these tapes, but the baseline must be independent of the hidden numerical seed labels. The query count includes all raw preprocessing and future queries.

E.3 Programming the challenge hash, including rejected attempts

Put $M_0 = 2^{256}$, the number of salts. For each salt ρ, the recognized complete challenge inputs containing ρ form a *slice* with projector P_ρ; the slices are disjoint, so $\sum_\rho P_\rho \le I$. At the first occurrence of each attempt name, we require the salt to be uniform and independent of the entire entry state and the challenge table.

The attempt forms its challenge input, which we call its *point*, before its single challenge query. The rest of the attempt may depend arbitrarily on that challenge, but makes no second challenge query before release. On rejection, the reset requirement stated above applies. All parties see one current prefix oracle; no second oracle exposes replaced values. Reprogramming the oracle replaces its value at a point. Replacements for successful attempts persist, and the last successful write to a point determines its value.

The comparison concerns one run of the experiment, from key generation to the final check, so its counts are not multiplied by the number of times later testers execute the run. In one run, only the adversary's h typed and r raw queries, the V verification requests, and the check of the final forgery can address a challenge input. The key, dither, matrix and message inputs differ from challenge inputs in length or tag. Private-tape inputs are disjoint from challenge inputs because role fields parse injectively, which is an assumption; the inputs for the private stream used only by the public sampler in the proof have 2060 bytes, so they differ from the 2916-byte challenge inputs by length; after the private streams are replaced, intended private reads address auxiliary streams indexed by key-generation call. Each of these accesses is one clean gate $|x, y, 0\rangle \mapsto |x, y \oplus f(x), 0\rangle$, however many underlying calls it makes, and on inactive branches its dummy fetch is computed and uncomputed against the same table. Changing the challenge answers on a set C of inputs therefore changes the gate by an operator of norm at most 2 supported on inputs in C. The $h + r$ adversarial gates may act on superpositions. Verification requests and the final forgery are classical in the security model of Section F.1. A gate with a classical input changes the state only when that input lies in C, and its trace-distance cost is the probability of that event. The analyzed Pretzel-(4,4) signer does not call the verifier before releasing a signature (Appendix D). A signing attempt makes at most one challenge query; if it makes one, that query is the one changed in this comparison.

First argument: one salt at a time. Consider the first occurrence of one attempt name, and call its salt ρ the reserved salt. During the *prefix*, the part of the run before the point is formed, *puncture* the whole slice of ρ, so that the prefix does not use the oracle's values on that slice. A superposed gate changes a purified state by at most $2\|P_\rho\psi\|$, and averaging over the independent salt bounds its squared mass by $1/M_0$. A classical gate, including an honest challenge read at another salt, changes the state with probability at most $1/M_0$. We first puncture the prefix of the real experiment. At a point that has not had its value replaced, the unused oracle value can then be read, or freshly sampled and installed, with exactly the same joint law of the current oracle. At a point whose value has already been replaced, this intermediate experiment uses the point's current value. The intermediate also installs the value when the attempt rejects; this oracle update on rejection exists only in the proof. Then remove the puncture from the prefix of the comparison experiment, which still carries the oracle update made on rejection.

To remove the oracle update made on rejection of a rejected attempt from the unpunctured comparison experiment, fix the base oracle and write the entry state $\sum_h |h\rangle\langle h| \otimes \sigma_h$ on its classical honest blocks. After averaging over the private fields of the failed attempt, its rejected state has the form

$$\begin{aligned}\sigma_\rho^\perp &= \sum_h f(h,\rho)|h,n,\perp\rangle\langle h,n,\perp| \otimes \sigma_h \\ &\le \tau := \sum_h |h,n,\perp\rangle\langle h,n,\perp| \otimes \sigma_h, \qquad 0 \le f \le 1.\end{aligned}$$

Here n is the attempt identifier, fixed before the reserved salt is drawn. The reset interface requires that the rejection marker contain neither that salt, nor the failed proposal, nor any private field that a later continuation can read. The outcome can still depend on the salt through $f(h,\rho)$; its probability is retained without conditioning. With this interface, τ is independent of the reserved salt. Later baseline CP maps preserve this operator inequality and cannot inspect failed fields, so their average query mass in the reserved slice is at most $1/M_0$. Over the rest of the run, removing the oracle update made on rejection therefore costs

at most $2/\sqrt{M_0}$ per superposed gate and $1/M_0$ per classical gate. Later successful writes can hide part of the support of the oracle update made on rejection but cannot enlarge it. Because this argument dominates the unnormalized rejected state, it does not need rejection to leave the oracle state unchanged.

Telescope over the N first occurrences, with exact repeats replaying their cached results. The time at which a point is formed is random, since it depends on retries and on adaptive scheduling. Telescope each puncturing so that the perturbation at every gate acts on the query state of its unpunctured reference circuit, namely the real experiment and the comparison experiment with its oracle update made on rejection; the removal of the oracle update made on rejection uses the continuation without the oracle update made on rejection, as above. These reference circuits agree in law before the point is formed. The punctured circuits need not, since puncturing can change earlier answers and hence the schedule. Suppose a superposed gate precedes the point with probability p in the reference circuits; before the point, their queried registers are independent of the reserved salt. By Jensen's inequality, the gate's cost is at most $4\sqrt{p/M_0}$ in the two puncturings and $2\sqrt{(1-p)/M_0}$ in the removal of the oracle update made on rejection. Since $4\sqrt{p} + 2\sqrt{1-p} \le 2\sqrt{5}$, each point charges each superposed gate at most $2\sqrt{5}/\sqrt{M_0}$. A classical gate costs at most $2/M_0$ per point. Honest reads at other first occurrences contribute at most $3\binom{N}{2}/M_0$ over all points. Finally, the public signing simulator samples fresh raw challenges even at points whose values were previously replaced. Before the first collision at which the experiments differ, the new salt of the jth first occurrence matches one of at most $j-1$ existing salts with probability at most $(j-1)/M_0$. Coupling and summation add $\binom{N}{2}/M_0$, which proves the bound $e_{C,1}$.

Second argument: all remaining salts at once. After the private streams are replaced, the salt of each first occurrence is read from an independent stream indexed by key-generation call and the other complete input fields that nothing else reads before that occurrence. Predrawing independent uniform salts $\rho_1, \ldots, \rho_N$ in order of first occurrence is therefore exact lazy sampling. Both experiments draw the same list, and conditioning both on distinct salts costs $\binom{N}{2}/M_0$ once. On the distinct branch no challenge input has its value replaced twice, so the simulator's fresh raw challenges agree with the intermediate experiment below. When c first occurrences have drawn their salts, puncture the union of the slices of the remaining salts, and telescope so that each perturbation acts on the state of the unpunctured reference run. Given the drawn salts, that reference state before the current gate does not depend on the remaining salts; the punctured state may. The remaining salts are distinct uniform values among the other $M_0 - c$, so the expected squared mass of the reference state in the union is at most $(N-c)/(M_0-c) \le N/M_0$. A superposed gate costs at most $2\sqrt{N/M_0}$, and a classical gate at most N/M_0. Honest challenge reads lie outside every punctured slice, because each attempt draws its salt before its read. As in the first argument, at each point the unused oracle value can now be read, or freshly sampled and installed, with exactly the same joint law of the current oracle, including the oracle update made on rejection. Removing the puncture from the comparison experiment, which still carries the oracle updates made on rejection, costs the same again, with the unpunctured comparison experiment as reference.

Then remove all oracle updates made on rejection from the unpunctured comparison experiment. For a rejected point k, fix all other salts. As above, its rejected state is dominated by an operator independent of ρ_k, and the later maps without oracle updates made on rejection depend on the other salts and the fixed oracle, not on ρ_k. Given the other salts, ρ_k is uniform on $M_0 - N + 1$ values. The squared mass in the union of the rejected slices is at most the sum of the individual masses, so a superposed gate costs at most $2\sqrt{N/(M_0 - N + 1)}$ and a classical gate at most $N/(M_0 - N + 1)$. On the distinct branch, honest reads of other points never address a rejected slice, and successful oracle updates are not changed. The three union steps and the distinctness cost prove the bound $e_{C,2}$.

Without using that verification requests and the forgery are classical, one can charge their $V + 1$ gates at the superposed rate; this gives both bounds with $h + r$ replaced by $h + r + V + 1$ and the terms in $V + 1$

removed. Charging every counted query in each of the three steps of the first argument instead gives the coarser bound $6NQ/\sqrt{M_0} + \binom{N}{2}/M_0$.

Attempt positions that are never reached act as the identity, and no entropy is drawn there. A failed call remains the product of its up-to-512 rejected subchannels; errors and successful attempt counts occur at their original positions. The final simulator replaces the oracle value at only the points of successful, returned signatures.

E.4 Implementation interface and order of setup replacements

The functional-correspondence assumption includes the following concrete requirements. All six stream constructors using encoded inputs, cached prefixes, cloned sponge states, and continuation offsets denote one persistent stream of the complete message for each exact name; role fields parse injectively; and the output length selects a view of that stream rather than a new name. The numerical PRF keys and key seeds, and any caches that depend on them, are held in protected registers and do not appear in the normalized active circuit except where streams are named. Equality of public fields does not imply equality of private PRF keys. The normalized circuit preserves the implementation's errors, order of entropy requests, stream reuse, and observable statistics. It has its own reversible implementation, which gives no inverse access to the implementation's secret caches. The absence of caches depending on numerical keys or seeds is a property of this normalized circuit and its chosen inverse; it does not mean that the Rust implementation caches no sponge states. The single-read, reset, and unread-row conditions stated earlier are required at the corresponding points of the proof. They are assumptions about the logical interface, supported by source inspection. They are not established by our numerical checkers, and they are not claims of physical erasure. The theorems do not cover an implementation that violates them.

The normalization can be written explicitly, with streams indexed by key-generation call. For each key-generation call i that the adversary may make, keep its numerical PRF key w_i in a protected register, and represent an intended stream by its role, i, the rest of its complete input, and its offset. Fix distinct labels. Swapping the full row of the ordinary oracle table at each protected name with an independent stream indexed by key-generation call and the other complete input fields preserves the joint uniform law of the table. Intended reads then use the independent stream directly, while external reads at the protected names see the swapped-in rows, which we call the replacement overlay. The observer has no separate access to the displaced rows. In the comparison baseline, remove the external overlay and keep the intended reads of independent streams. The baseline's active state and the order of its operations are then independent of the numerical labels, as the hidden-support lemma requires. The chosen inverse undoes this representation indexed by key-generation call and its retained history; it never reconstructs a physical sponge cache.

This independence persists through the later hybrids. The PRF key is drawn independently of the core key and of the randomness of the dither selection. The dither selection reads only public expansions. The stream monitors, transport, caps, and saved retry outcomes depend on the contents of independent streams, but not on the numerical label. Successful replacements of challenge values are indexed by complete public challenge inputs, and saved rejection outcomes satisfy the reset requirement. The private streams used only by the public sampler in the proof belong to the family indexed by PRF keys and are covered by the same replacement. For the key-seed domain, intended CBD reads use cached rows indexed by the candidate. The trits, the key-norm acceptance decision, and setup metadata may depend on those rows without depending on their numerical seed labels. After the steps that make signing witness-free and introduce the second, independent key-norm search (both below), no active operation reads the numerical labels. The final experiment, in which streams are indexed by key-generation call, is therefore independent of whether the labels were distinct. These statements rely on the functional correspondence above, not on physical erasure.

After the finite-to-ideal transport comparison and the selective-answer comparison, the identity of all classical outputs and retained quantum states makes signing independent of the witness. The public samplers

are then extended to every canonical core key, before the key-norm search is replaced by its first CBD draw without conditioning. Couple the first draw of the two experiments; they agree when it passes, which costs at most η per key-generation call, and the other branch retains all capped failures. No theorem about private keys outside the key-norm range is used. The sharper factor in the bound comparing real and simulated signing follows by keeping a second, independent search for a key that meets the key-norm restriction, whose candidates determine only the loop's control information and are never used as the key. Here η is the exact probability that one independent CBD candidate fails the key-norm restriction, with $\eta < 2^{-192}$, rather than a numerical upper bound substituted for it. Conditional on the index j of the first passing candidate, let L_j be the law of the permitted loop metadata, and let ν_A^+ and ν_A be the laws of the CBD core key with and without conditioning on the key-norm restriction. They have total variation at most η. The map producing the successful classical outputs and retained quantum state is

$$\sum_{j=1}^{512} \eta^{j-1}(1-\eta) \int L_j(d\ell) \sum_b \nu_A^+(b)\, C_{j,\ell,b}(\sigma_E).$$

Replace only ν_A^+ by ν_A. The common CPTP continuation includes encoding the core key, selecting the first candidate whose bounded dither expansion succeeds for Pretzel-(4,4) or Pretzel-(5,5), retaining exhaustion, and all future interaction. The expansion bounds are those of Appendix D.2 and Section I.1, respectively. This map needs neither a witness nor a row-quality condition. At rank four it also applies to matrices outside G of Lemma D.3. The map for the event that all 512 candidates fail has mass η^{512} and is the same in both experiments. Contractivity gives $(1-\eta^{512})\eta$ per invocation.

For both parameter sets, the selection reads only the core key and public setup coins, returns its selected seed or failure, and never resamples the external core. The continuation is therefore the same map on both sides for every canonical b. At level five the resulting factor is $(1-\eta_{55}^{512})\eta_{55}$.

This factorization requires that the fields of the accepted witness are not kept as metadata that later steps can read; isolating the key-seed domain and making the continuation witness-free ensure this at the relevant point. In particular, the proof never installs an external core key while an accepted witness is still readable. Candidates of the second search determine only the loop; they are never passed to secret-key preparation as a fabricated witness. The dither selection runs once, including its possible failure, and never restarts or resamples the external hard instance.

In this witness-free experiment, the predrawn public matrix seeds are independent of the fresh matrix oracle and of the external Module-LWE sample. Install the external uniform matrix at the predrawn seed before any charged preprocessing. Averaging over the external matrix then gives exactly a uniform matrix table, including all seed collisions and unused positions in the predrawn seed list. This argument fails for a seed chosen adaptively after inspecting the fresh table; the order of replacements established above, from the implementation to the public experiment, is what permits the independent seed draw here.

Finally, once no active operation still reads numerical labels, index the private streams by key-generation call instead of by numerical label. By convexity, the single global event that the labels are not distinct is charged once; neither the external Module-LWE instance nor successful signing is conditioned on that event. At the first stream-monitor overrun, stop both experiments and allow arbitrary continuation; charge the overrun probability once. The sampler bounds and finite oracle-simulation bounds are those specified in the theorem for the parameter set in use.

Applying measure-and-reprogram. Apply the single-input theorem of Don, Fehr, and Majenz [13, Section 3, Lemma 1 and Theorem 2] to the complete challenge circuit, with its fixed worst-case budget of q_B challenge queries. The theorem's first-stage output fixes the typed challenge input. Verification with the challenge given instead of hashed checks that the continuation reconstructs that same input, in addition to canonical parsing, eligibility, and the stated verification checks; it does not obtain a replacement challenge

by querying the oracle. Let $\epsilon_0 = [\epsilon - \Delta_F - \Delta_\mu - P_{\text{pair}}]_+$ be the lower bound on the unconditional probability of a good output in the experiment with the public simulator. The exact branch experiment then succeeds with probability at least $\epsilon_0/(2q_B + 1)^2$. Our finite branch sampler selects each of its $2q_B + 1$ branches with probability $\lambda_{\text{MR}}/(2q_B + 1)$ and sends exhaustion to the common output with zero success. Its unconditional success probability is therefore at least $\lambda_{\text{MR}}\epsilon_0/(2q_B + 1)^2$. Neither the output branch nor verification with the given challenge adds an uncharged oracle query, and the argument does not condition on the sampler's success.

F Finite QROM Reduction

We analyze the Pretzel-(4,4) construction of Section 4. The bound depends on the sampler's complete behavior, including aborts and retries, and is conditional on the implementation corresponding to the specification, as stated in Assumption F.1.

F.1 Security model

This subsection specifies the security experiment and the parameters that bound the adversary's resources.

The experiment uses honestly generated keys that never change, a fixed honest application context, and classical key-generation and signing requests. A *key handle* identifies a stored key. Hashing is modeled by an ideal extendable-output function (XOF): each complete input byte string determines a persistent random output stream, and a query requests a prefix of that stream. The adversary may query this function in superposition, and all its queries are charged.

Public verification and preparation requests are classical API calls, like key-generation and signing requests: the experiment measures each request before answering it. An adversary that wants a superposed verification can run the public verifier itself; its challenge queries are then charged as direct or raw queries, defined below. The final forgery is a classical output. Besides API results, the adversary may observe the declared statistics, including setup failure, signing failure, the 512-attempt cap, successful attempt counts, and repeated requests.

An attempt's *name* is its complete encoded input to the private randomness streams. Attempts with equal names, even in different calls, reuse the same random stream and cached result. Forgery is defined with respect to canonical public keys and the log of returned signatures. Ordinary unforgeability (EUF) requires a forgery on an unsigned message; strong unforgeability (SUF) also counts a new signature on a previously signed message.

The adversary may start with quantum advice, which is independent of the newly sampled oracle tables and experiment coins. All oracle-dependent preprocessing runs inside the experiment and is charged. The reduction inherits the adversary's advice class and storage. Hardness against uniform quantum polynomial-time (QPT) algorithms therefore applies only to efficiently preparable advice, with its preparation charged; arbitrary independent quantum advice requires the corresponding hardness class.

The experiment excludes secret-key export, corruption, private contexts modified by the caller, private inverse access, entropy faults, and physical leakage. We assume ordinary functional semantics for integers, buffers, and the foreign-function interface (FFI). This analysis does not cover compiler correctness or the concrete SHAKE instantiation.

The experiment uses the Pretzel-(4,4) finite sampler and release rule specified in Appendix D; the release rule returns an attempt's signature only after all of its accumulated abort and public-cap checks pass, the response fits in the encoding, and encoding succeeds, without first running the verifier; an encoding failure rejects the attempt. Let Γ denote the joint law of the matrix and raw streams induced by the rejection sampler with encoded hash inputs of Section 4.1. The public matrix and its raw stream follow this same law, since

the inputs are relabeled by a bijection. The experiment uses a functional matrix interface, which returns the matrix directly and answers bounded raw-prefix queries according to this joint law. Write $A_h = [-b \mid A_0]$ and $H = [A_h \mid I_4]$.

Let $1 \le K \le 2^{64}$ bound the number of potential *setup positions*: one for each key-generation call the adversary may make, including calls that fail and positions that are never reached. Let S, V, h, r bound, respectively, the numbers of signing requests, public verification or preparation requests, direct typed-challenge queries (which return a folded challenge), and raw-prefix queries (which return a prefix of a raw output stream). Direct and raw queries are counted separately. At most S signing requests entail at most $A_{\text{att}} = 512S$ signing attempts. Let $N \le A_{\text{att}}$ bound the number of distinct attempts, identified by their complete private-stream inputs.

Sampling and implementation conditions. The analysis uses the 1361-byte Pretzel-(4,4) encoding of Section 4.5. Applying it to the implementation requires bounds on the complete sampling process, including failed attempts and retries, and correspondence with the specified setup, signing, and oracle interfaces. Throughout, agreement on successful signatures alone is not enough.

Assumption F.1 (Finite sampler and implementation model)**.** The Pretzel-(4,4) implementation satisfies the following conditions.

1. Its finite proposal and transport sampler implements the private-stream schedule and probability precision of Appendix D. The bounds stated in Section F.3 on its complete output distribution, including aborts, and on the monitor hold on every permitted history at the start of an attempt, including the stated dependence between shared transport coins.

2. Its encoded oracle inputs (names) implement the specified parameter set, field lengths, and role separation. The input encoding is injective, and the query and byte bounds of Section F.2 hold for the complete interfaces, including cleanup and retained state.

3. The check that the response fits in the encoding and the encoding itself are deterministic functions of (z_h, U, r_e) and public data, evaluated after the last read of private randomness. Both sides of each comparison of complete output distributions, including aborts, apply this same public map and retain the complementary abort mass.

4. Setup implements the candidate order, the first successful bounded dither expansion, and the failure interface of Appendix D. The implementation and ideal sides of each comparison use that same public computation, under the requirements on subroutine behavior and exclusive use of temporary storage: every subroutine that the setup and signing loops call (dither expansion, stream construction, one complete attempt) is a deterministic function of its stated inputs with no hidden side effects, and each scratch buffer, stream, and cache is used exclusively by one computation and is returned in its specified cleared state. These are logical contracts of the security experiment; they do not assert physical erasure of memory, whose status Section 7.2 describes.

5. The requirements on complete outputs, resetting state after failure, replaying repeated attempts, and the other functional requirements of Appendix F and Appendix E hold on every permitted history, including failures and live updates of the oracle simulators and the monitor. In particular, the equality of retained outputs required below includes continuation state.

6. Matrix operations use the interface defined above, with joint law Γ, preserving all allowed outputs, failures and state retained for later calls. The finite query, work and storage bounds cover this interface and its simulation in Section F.2. The implementation's matrix sampling loop has no fixed limit

on retries. Its running time and number of stream reads are excluded from both the experiment's observations and these bounds.

Theorem 5.6 assumes all the conditions above. Numerical certificates and proofs about the setup and signing loops do not establish that the compiled implementation satisfies the full assumption.

For Pretzel-(5,5), the setup clause of Assumption F.1 uses the candidate order, bounded expansion and failure interface of Section I.1, with no public-row generation or cache. The comparisons use this same computation on both sides. Section I.2 counts its queries and work.

F.2 Oracle query counts and finite simulation

An *oracle simulator* is a finite circuit that answers a bounded number of oracle queries, with private state that persists through forward and reverse executions.

Honest query counts. For each honest stream, we charge a complete bounded-prefix query and its uncomputation. Rank-four setup uses at most 512 key-seed streams and 2048 dither candidates, with no public-sample streams. Keeping the twelve additional inputs for matrix, integrity and selected-dither work gives 2572 inputs and 5144 calls including cleanup. The allowance $7200K$ bounds this count at every hybrid. A signing attempt uses at most six private streams and one challenge input, which 16 calls cover. Message hashing costs two calls per signing request. Verification with a key that has not yet been unpacked uses at most two matrix streams, one dither, one message, and one challenge input, which 12 calls per request cover. Matrix work is also charged separately, which is conservative. Honest private preparation runs once per immutable key handle and is cached; later public preparation is charged to V.

Lemma F.2 (Caching and challenge programming)**.** The hash inputs used to prepare rank-four keys, dithers, the matrix and the private streams lie outside the challenge domain. Programming the challenge oracle therefore leaves every cached value and cache entry unchanged, so caching does not affect any challenge-programming hybrid. The other replacements are fixed before execution and are treated in Appendix E.

Proof. The key-seed, matrix and dither inputs carry their own domain prefixes, so none coincides with a challenge input (Appendix D). Each private role is encoded separately from the challenge role, so none of these inputs is a challenge input. The untagged input, the public-body integrity hash, has 1728 bytes, while every challenge input has 2916 bytes. Hence each cached computation reads only oracle entries on which the original and the programmed oracles agree, and induction over the cached computations, in the order they are made, shows that each produces the same value and the same cache entry in both; a cache hit replays that value. □

A signing attempt can query the challenge oracle, so attempt outcomes are handled by a replay rule instead: each hybrid records the complete outcome of an attempt, success or rejection, at the first occurrence of its name, and a repeated name replays that cached result without a new challenge query. Monitor and oracle simulator state advance by the cached result's update; no saved state is restored. This is the exact-name replay interface used by the comparison of real and simulated signing. Answers to verification queries are not cached; they keep their recorded version and are reversed with it, as described below.

Total query bounds. Assumption F.1 requires the following query bounds, which include honest work and inverses. They must hold for the implementation's packed-slot scratch and stream interfaces and for the intermediate comparison circuits.

$$R = r + 7200K + 16A_{\mathrm{att}} + 12V + 2S + 8, \qquad D_M = 8K + 4(V + 1),$$

$$
\begin{aligned}
F_H &= 16R + 2h + 2,\\
F_U &= 44R + 2A_{\mathrm{att}},\\
F_M &= 32R + 2D_M,\\
Q_{\mathrm{bin}} &= 180R + 6h + 8 + 6A_{\mathrm{att}},\\
Q_{\mathrm{mat}} &= 96R + 6D_M,\\
Q &= Q_{\mathrm{bin}} + Q_{\mathrm{mat}}.
\end{aligned}
$$

Here $Q_{\mathrm{bin}} = 3(F_H + F_U) + 2$ and $Q_{\mathrm{mat}} = 3F_M$. Each triple below lists the numbers of challenge, auxiliary-binary, and matrix calls; the last triple gives the coefficients of R in F_H, F_U, and F_M. A complete raw gate costs $(2, 5, 2)$. A *clean replacement gate* first computes the underlying answer into scratch, selects the replacement, XORs it into the caller's answer register, and then reverses the computation. It makes two underlying calls, even on inactive branches, and does not assume free controlled oracle access. The successive replacements therefore scale the counts as follows. Replacing cached key rows doubles them. Replacing the private family (the private streams of each attempt) doubles them and adds two auxiliary calls to obtain and clear its independent row. Replacing saved successful outcomes doubles the result, giving $(16, 44, 16)$. Replacing the matrix at the predrawn seed of the external instance doubles the matrix component, giving $(16, 44, 32)$. Direct matrix calls contribute $2D_M$. A direct challenge query with the programmed value is charged two challenge calls, which explains the term $2h + 2$. Public-sampler tape reads and cleanup contribute $2A_{\mathrm{att}}$. Allowing three executions, as explained below, then gives the oracle simulator totals displayed above.

Correspondence with the implementation. These counts refer to the normalized oracle circuits. To identify the implementation with them, express its matrix operations through the functional interface above, and fix the underlying oracle table T and all input coins. The required functional refinement preserves the complete retained output $f_T(x)$, including abort and continuation state, for every table T and every bounded input x outside the common monitor-overrun event. This event arises because the implementation's leading-sampler loop has no trial cap. The analysis monitors this loop with a fixed byte limit; if the limit is exceeded (an *overrun*), both sides return the same failure output, and the comparison of real and simulated signing charges this event once, as $A_{\mathrm{att}}\tau$. Reversible computation, XOR of the output, and uncomputation against the same table implement

$$|x, y, 0\rangle \longmapsto |x, y \oplus f_T(x), 0\rangle.$$

Pointwise equality of f_T therefore implies that these clean circuits are equal on arbitrary superpositions, including with reference systems. Saved interaction histories are part of the input and output, and the underlying table stays fixed during uncomputation. The reduction builds this circuit itself; it does not invert the erase instructions of the compiled signer. The correspondence between implementation and specification must still establish equality of the complete retained output.

The implementation's matrix expansion, in contrast, reads rejection streams until both matrix streams are complete, with no trial cap, and the finite prefix width below does not bound its running time. The theorem therefore covers only implementations that obtain the matrix through the functional interface, as Assumption F.1 requires. The leading-sampler monitor and the matrix simulator's coefficient-sampling error do not supply this term: matrix expansion also occurs during key generation, including when $S = 0$ and hence $A_{\mathrm{att}}\tau = 0$.

For rank-four setup, fixing the public core, oracle and candidate-start seed fixes every candidate expansion and the first successful candidate or exhaustion. Induction over the bounded candidate list therefore preserves errors, entropy order and cleanup under the stated functional contracts; no row cache is used. Signing retains its exact-name replay contract. Equality of the complete retained output additionally requires the attempt arithmetic, stream laws, and coherent interface specified here. Assumption F.1 includes these obligations.

The challenge circuit. The F bounds cover the intermediate comparison circuits. Once signing has been replaced by public simulation, the private-stream replacements are no longer active. Each execution of the resulting challenge circuit or of its adjoint then makes at most q_B challenge queries, where

$$q_B = 2h + r + V.$$

Every challenge access in this circuit uses the following gate, which makes exactly one query.

1. Without a challenge query, compute the activity bit, the requested prefix mask m, and the last saved successful outcome with the requested input, with its hit bit and value v.

2. An active access without a hit queries its challenge input under mask m. Every other branch, including inactive, hit, and nonchallenge branches, queries the oracle under mask zero at a fixed dummy input $w_\perp$ that is never a challenge input. To apply the mask, swap each answer bit whose mask bit is zero with an ancilla qubit in state $|+\rangle$, apply the oracle once, and swap the bits back. Since $X^b|+\rangle = |+\rangle$, the ancillas absorb these answer bits and return unchanged.

3. On an active hit, XOR v under mask m. Finally, uncompute the controls.

Here the value queried at a challenge input is its 256-bit raw label, as specified under consistent raw and typed access below (Section F.2). The gate equals the raw oracle with programmed values, masks, and controls on every input, including a programmed value in superposition, and is its own inverse. Its 256 ancilla qubits and their 256 Hadamard gates are charged. A raw query uses one such gate. A direct typed query returns the fold of the label: it applies the gate with the full mask to a zero scratch register, XORs the fold of the scratch into the answer register, and applies the gate again, so it uses two queries.

Verification requests. A verification request computes the challenge of its measured request with one such gate and computes its verdict. It then keeps the answer in a workspace register s instead of uncomputing it. The kept answer is a copy controlled by the saved request, by the list of programmed inputs and values present at that time, and by the fixed table. Input–value pairs are only appended, and a value field changes only when reverse execution removes its pair. Every later operation before the matching adjoint acts on registers disjoint from s and is block diagonal in these controls. These operations include the acceptance projectors, the dephasing Δ_ξ, response measurements, the filter of Lemma C.2, and the lift adapter. No operation measures, resets, or discards s, and only the matching adjoint removes it, with one query. The copy therefore commutes with all these operations and cancels in the adjoint. Hence acceptance probabilities, pulled-back projectors, and the return probabilities of Appendix C.2 are unchanged. This argument requires the request to be measured; for a superposed request, the kept copy can change these quantities.

The final forgery check. Checking the final forgery makes no challenge query. This check is the output branch of the measure-and-reprogram theorem [13], which evaluates acceptance at the supplied base challenge value. Its challenge input, the value hashed to obtain its challenge, is computed from the output using only message queries and arithmetic. The good event that we count requires that this input was never programmed, so the base and programmed values agree there.

Algorithms that run the challenge circuit. The extractor and the testers run this same circuit. Public sampling, preparation that involves no challenge, and setting the chosen hash values make no challenge query. The public setups of all potential positions involve no challenge and are cached. Verification with the challenge given instead of hashed, and saved response reconstructions, use arithmetic but no additional oracle calls, and they keep Pretzel's order of verification with the challenge given instead of hashed. The

extractor and the chosen measurement-hiding tester make at most the calls of three executions; confirming a collision adds at most two ordinary calls. The larger bound Q remains valid at every intermediate hybrid and for all complete algorithms that test the lattice assumptions. Neither K nor κ counts repeated executions of one tester.

Input and output widths. Let I_{adv} and L_{adv} bound the lengths of the adversary's oracle inputs and requested outputs, and let $C_{\max}$ and $M_{\max}$ bound context and message lengths, all in bytes. Let L_{lead} be the byte limit of the leading-sampler monitor. Define the byte bounds

$$I_* = \max(I_{\mathrm{adv}}, 4096 + C_{\max} + M_{\max}),$$
$$L_* = \max(L_{\mathrm{adv}}, 605{,}292{,}544, L_{\mathrm{lead}}).$$

We require $I_* < 2^{64}$, and we require the 64-bit length fields of the implementation and of the proof's adapters to be representable. The requested output length masks a fixed full-length value and is not part of the name. Set

$$I = 8(I_* + 9), \qquad L = 8\left\lceil \frac{13\lceil 8L_*/13 \rceil}{8} \right\rceil.$$

The extra rounding supplies complete 13-bit words for the construction of the joint matrix and raw-stream distribution Γ: 605,292,544 raw bytes require 605,292,545 bytes from the base oracle simulator. The leading-sampler monitor, with its limit of 318 trials, fits within this fixed width.

Consistent raw and typed access. A matrix seed identifies both a SHAKE output stream and the matrix decoded from it. A simulator must answer queries to either representation consistently. The construction below embeds a supplied matrix into a raw stream with the correct conditional distribution.

Use disjoint auxiliary namespaces for ordinary raw inputs and for matrix rejection patterns. For each matrix seed and stream, split an independent auxiliary string into consecutive 13-bit words. Keep the words of value at least 7681. Replace the first 2048 words below 7681 by the successive matrix coordinates assigned to that stream, in the order of Appendix D, and leave later words unchanged. Conditional on the pattern of accepted and rejected words, the rejected values are independent and uniform in $\{7681, \ldots, 8191\}$, and the substituted coordinates and later accepted words are independent and uniform in $\{0, \ldots, 7680\}$. Thus the resulting raw stream is uniform when the matrix is uniform, and its first 2048 accepted values are exactly that part of the matrix assigned to that stream. The two streams use disjoint coordinates. This proves the joint law Γ, including for prefixes that end before the matrix is complete or within a 13-bit word. With an externally supplied matrix, it gives the corresponding conditional law of the stream.

For a recognized challenge input, represent its typed challenge by the unique 255-bit preimage under the specified fold. Return these preimage bits in raw positions 0 through 254, and independent auxiliary bits from position 255 onward. The last bit of the first 32 bytes is therefore independent of the challenge. Equivalently, the oracle simulator may keep a uniform 256-bit hash label and fold it for typed access; both representations give the same joint law of raw output and challenge. Other encoded inputs use their own ordinary auxiliary string. Parsing and requested-length masks preserve complete names.

To implement this coherently, fetch the ordinary auxiliary value, rejection pattern, matrix, and typed challenge into zero scratch registers. Select the appropriate answer, XOR its requested prefix into the caller's answer register, whose contents are arbitrary, and uncompute everything. Inactive branches use dummy addresses. The gate uses $(2, 4, 2)$ calls, within the $(2, 5, 2)$ allowance above. Equality on every basis state of the input and answer registers extends linearly to any state of the client and reference systems.

Replacing private and key-seed streams. The names of a key's private streams contain its private 32-byte PRF key. Replacing these streams parses the role and the PRF key, then reads an independent auxiliary row indexed by the key-generation call and all other encoded fields. The role and every field remain part of the name. A one-byte namespace tag and the replacement of a 32-byte PRF key by an eight-byte call index fit within the input width above. Key-seed replacements use independent cached full rows; when PRF keys coincide, the first call's streams are used. Appendix E gives the coordinate-swap argument for uniform tables and the hidden-support argument.

Programming the challenge oracle. Challenge replacements for successful attempts use an ordered private list of complete encoded inputs and their 256-bit replacement values. The last pair matching the queried input determines the current prefix; the raw suffix is unchanged. Setting a new value appends a pair without reading the old challenge value. Query gates preserve the list, and reverse execution undoes appends in reverse order. Failed attempts append no pair. The clean selection construction above gives the stated counts for these wrappers. The adversary has no separate interface to displaced values or to the underlying oracle simulators before programming. These explicit circuits show that the oracles can be realized; the comparison of real and simulated signing still requires its stated freshness, naming, and reset hypotheses.

The matrix oracle simulator. For the matrix oracle simulator, choose the degree-20 extension field $\mathbb{F}_{7681}[X]/(f)$ and use 205 independent polynomials of degree $2Q_{\mathrm{mat}} - 1$. Each value expands to 20 residues; discard four of the 4100 residues. The remaining 4096 residues have exactly the required joint $2Q_{\mathrm{mat}}$-point law. The field contains an injective encoding of every 256-bit seed, since $7681 > 15 \cdot 2^9$ and $15^5 = 759{,}375 > 2^{19}$ imply $7681^{20} > 2^{256}$.

For any $s \le 2Q_{\mathrm{mat}}$ distinct encoded seeds, evaluating one polynomial is a surjective linear map from its $2Q_{\mathrm{mat}}$ coefficients onto s field values, because the first s columns of the evaluation matrix form an invertible Vandermonde matrix. Uniform independent coefficients therefore give uniform independent values at these seeds. Expansion in the fixed field basis, independence of the 205 polynomials, and discarding the unused coordinates preserve the required joint law.

We use the fixed polynomial $f(X) = X^{20} + X + 11$, so no polynomial search is needed. Exact modular arithmetic verifies

$$X^{7681^{20}} \equiv X \pmod{f}, \qquad \gcd(f, X^{7681^{10}} - X) = \gcd(f, X^{7681^4} - X) = 1$$

using explicit Bézout identities for both coprimality checks. The first congruence implies that f is square-free and that the degree of each irreducible factor divides 20. The two coprimality checks rule out every proper divisor $1, 2, 4, 5, 10$ as a factor degree, which proves that f is irreducible. Field elements are represented by twenty residues. A multiplication is a convolution followed by descending reduction with $X^{20} = -X - 11$, which takes 400 base-field multiplications and at most 38 reduction updates. Horner evaluation of the 205 polynomials uses at most $205(2Q_{\mathrm{mat}} - 1)$ field multiplications and as many field additions, followed by the charged reversible cleanup.

The binary oracle simulator. For the binary oracle simulator, use a private sparse Fourier table, following the Fourier-recording perspective of Zhandry [45]. Encode a name x of length $j \le I$ injectively as $0^{I-j}\|1\|x$, and let $T = Q_{\mathrm{bin}}$. In the Fourier basis, the initial table is zero. A query Hadamard-transforms the answer register, XORs the resulting Fourier answer label into the table value at x, and Hadamard-transforms the answer back. The update acts coherently on both registers; it is not a classical dictionary.

After j queries, each table basis state has at most j nonzero entries. Represent it by a canonical sorted private list with space for T entries, bits indicating which entries are used, and zero padding. A query inserts, XORs into, or deletes the addressed entry. Inserting a new entry into a full list, and any invalid encoding,

act as the identity, so the update is a total involution; within the stated query limit, no new query reaches the full-list insertion branch. The isometry from canonical lists to complete Fourier tables intertwines every reachable query. Hence the entire channel seen by the client and reference systems is exact, including arbitrary destination registers. A prefix query masks only the update; its length is not added to the name. The database persists across inverses, replay, and final tests, and it is never measured, exported, reset, or read for any other decision of the client.

A full scan and a stable one-position shift give a clean reversible update with $O(T(I + L))$ work. Computing and uncomputing with the total involution removes scratch. Including the answer Hadamards, the total binary work is $O(T^2(I + L) + TL)$ and the workspace is $O(T(I + L))$, without QRAM. These are exact Hadamard and Toffoli circuits: they need no search for a binary field polynomial and incur no rotation-angle error.

Approximation error. The matrix oracle simulator has $n_c = 8200Q_{\mathrm{mat}}$ base-field coefficients. Sampling each with $t_c \geq 1$ bounded 13-bit rejection trials and a fixed fallback gives the only approximation term of the oracle simulators, the oracle-simulation error

$$\delta_{\mathrm{sim}} = \delta_M = \min\{1, n_c(511/8192)^{t_c}\}.$$

For integer $\lambda_c \geq 1$, taking $t_c \geq \lceil(\lambda_c + \lceil\log_2 n_c\rceil)/4\rceil$ makes this at most $2^{-\lambda_c}$. The fallbacks are finite failure branches, not postselection. An oracle simulator whose query bound is zero is omitted.

Why oracle-relative hardness is unnecessary here. The proof first establishes measure-and-reprogram and extraction in the ideal QROM, and only then substitutes oracle simulators inside each entire constructed tester. Conditional on the matrix oracle simulator and the external instance, the binary simulation is exact. For the matrix oracle, the quantum polynomial method shows that the entries of the final density matrix have degree at most $2Q_{\mathrm{mat}}$ in the indicators of oracle values. The oracle simulator built from independent polynomials therefore supplies the required point law, including the client's reference system. Its coefficients are used only for clean evaluation and never for another decision, and inverses reuse the same coefficients. Substituting the oracle simulator changes one output probability by at most δ_M, and a distinguishing advantage by at most $2\delta_M$. Matrix evaluation keeps the reversible Horner cost of the 205 polynomials, and parsing, masks, linear scans of saved input–value pairs, and public arithmetic are charged. Although finite, the cost and storage of the polynomial oracle simulator can be enormous.

F.3 Comparing real and simulated signing, including aborts

Write $(m)_j = m(m-1)\cdots(m-j+1)$ and set

$$\begin{aligned}
\delta_G &= 1 - (2^{256})_K/(2^{256})^K, & R_K &= 512K,\\
\delta_K &= 1 - (2^{256})_{R_K}/(2^{256})^{R_K},\\
e_P &= 4KQ^2/2^{256},\\
e_{KD} &= \delta_K + (1-\delta_K)\min\{1, 4R_KQ^2/2^{256}\},\\
e_C &= \min\{1, e_{C,1}, e_{C,2}\},
\end{aligned}$$

where

$$\begin{aligned}
e_{C,1} &= 2\sqrt{5}\,N(h+r)/2^{128} + \big(2N(V+1) + 4\tbinom{N}{2}\big)/2^{256},\\
e_{C,2} &= 6(h+r)\sqrt{N/(2^{256}-N)} + 3N(V+1)/(2^{256}-N) + \tbinom{N}{2}/2^{256},
\end{aligned}$$

with $e_{C,2} = 1$ if $N \geq 2^{256}$. Appendix E derives both terms from the challenge accesses of one run. Here δ_G and δ_K are the probabilities that K and R_K independent uniform 256-bit labels are not all distinct. In the proof below, e_P pays for replacing the private streams, including the private stream used only by the public sampler in the proof, and e_{KD} for the key-seed streams. The term e_C pays for programming the challenge hash. Use the trivial distance one for impossible partitions, and clip distances when useful.

Write τ for the probability that the common stream monitor overruns, D_{core} for the per-attempt error between the complete finite and ideal output distributions, including aborts, e_{pub} for the approximation error of the public response sampler, and η for the probability that the key-norm restriction (3) rejects one candidate key. For the implementation, this comparison uses the bounds

$$\tau < 2^{-256}, \quad D_{\text{core}} < 5{,}103{,}841\, 2^{-192}, \quad e_{\text{pub}} < 2^{-297}, \quad \eta < 2^{-192}.$$

The setup experiment handles the fixed public dither, so D_{core} has no per-attempt term for a fresh dither. Lemma D.3 bounds public-filter rejection for matrices satisfying G, with exceptional probability ε_G in its ideal matrix model. This availability bound is distinct from D_{core}, the error between complete finite and ideal laws. Proposal and transport errors use the key-norm range and fresh private coins, not the dither selection rule; for each fixed dither the same deterministic public filter contracts their distance (Proposition D.4). The spread bound does not prove the numerical sampler assumption, which is required for every permitted history, including matrices outside G.

These numerical bounds are assumptions of the comparison for the Pretzel-(4,4) encoding as implemented. The numerical certificates supply component error bounds; applying them to the complete sampler and its stream interfaces requires Assumption F.1. The finite-sampler calculation combines 22 error terms and compares the coarse evaluator, the low-precision comparison that decides most transport steps, with the full correction. The joint transport calculation certifies containment margins for predicates that share one continuous coin; finite-precision errors are charged separately. Applying these calculations requires the complete output distribution, including aborts, of the implemented sampler, including its transport coins with a shared tail. The assumption concerns these shared-tail coins and the enabled compile-time features of the implemented build. The per-attempt comparison must hold on every still-related permitted history before the current private coins are sampled. A coarse evaluator may change which decisions are deferred, provided that its uniform error bound, causal coin dependencies, error allowance for exceeding the storage limit for deferred decisions, and full deferred verification establish the same comparison of completed output distributions, including aborts.

Assumption F.3 (Numerical sampler assumptions)**.** The security theorems for the three parameter sets take the following finite-sampler bounds as assumptions, on every still-related permitted history and with aborts counted as outputs. For Pretzel-(4,4): $\tau < 2^{-256}$, $D_{\text{core}} < 5{,}103{,}841\, 2^{-192}$ and $e_{\text{pub}} < 2^{-297}$, as above. For Pretzel-(5,5): the bounds of Section I.2, including $D_{55} < 2^{-304}$. For Pretzel-(3,3): $D_{\text{attempt}} < 2^{-169}$ of Section H.1, which in addition assumes the shared-tail replacement stated there. Applying these bounds to an implementation also requires Assumption F.1. The code that computes these bounds is published with the reference implementation.

Every rank-four key hybrid returns the first candidate whose bounded expansion succeeds, with the same exhaustion outcome. This is a public map on every canonical core, so it also applies after replacing the key column by a uniform one. The proof neither conditions on setup success nor discards matrices failing Lemma D.3. That lemma supplies row quality for availability; the comparisons of complete outputs below need no lower bound on acceptance. By Proposition D.4 and the lift of Appendix B.5, checking that the response fits in the encoding and then encoding it canonically adds no cryptographic error and preserves the response-measurement and distinct-pair arguments. The distance between real and simulated signing is

bounded by

$$\Delta_F = \min\{1, \delta_G + (1-\delta_G)\min\{1, e_P + A_{\text{att}}\tau + A_{\text{att}}D_{\text{core}} + e_C + A_{\text{att}}e_{\text{pub}} + e_{KD} + K\eta(1-\eta^{512})\}\}. \tag{F.1}$$

Proof of the comparison of real and simulated signing. Predraw the PRF keys of all potential setup positions, and condition the comparisons on their being distinct. Conditioned on this event, the labels are sampled without replacement. Isolate the entire private family, with all its roles, including the private stream used only by the public sampler in the proof, at cost e_P. A common flag marking the first overrun adds $A_{\text{att}}\tau$ once. Coupling each finite attempt with its ideal counterpart, without conditioning, adds $A_{\text{att}}D_{\text{core}}$; the ideal side is not conditioned on the coupling succeeding. Total arithmetic on the same events and deterministic public postprocessing preserve Pretzel's actual response caps and canonical encoding.

Under the exact-name replay interface of Section F.2 and the freshness and logical-reset conditions of Appendix E, that appendix proves that challenge programming costs e_C and that ideal transport and ideal public sampling have identical complete outputs, including complementary abort mass and the quantum state retained at the start of the attempt (the entry reference).

Here e_{pub} bounds the error in the joint distribution of response and gate for every supplied challenge, using fresh coins independent of the entry reference. Applying the same public caps and encoding preserves this bound for the complete output, including aborts, without conditioning on acceptance. Exact-name repeats reuse the cached result and incur no new sampling error. The public sampler's finite CDF sampler therefore adds $A_{\text{att}}e_{\text{pub}}$. It reads 92,264 bytes from an independent stream that exists only in the proof: an additional private stream beyond the implementation's six streams (salt, lead, secret, error, transport, and dummy) of Appendix D. This stream has a 2060-byte input and is distinct from those six. Its rows are replaced in the private-family step charged by e_P, so the sampler's reads need no further replacement. For a conservative circuit bound, each coordinate fully scans its table. The four tables, for even and odd leading coordinates and for secret and error coordinates, have 155, 154, 2129, and 2711 states (5149 in total). Charging 155 states to each leading coordinate gives $256 \cdot 155 + 1024 \cdot 2129 + 1024 \cdot 2711 = 4{,}995{,}840$ 320-bit threshold comparisons per sample, plus the gate comparison.

Replace the 512-byte key-seed rows, read at 53-byte encoded inputs, by independent uniform candidate rows, then apply the second, independent search for a key that meets the key-norm restriction, early matrix-embedding and indexing of private streams by key-generation call instead of by PRF key of Appendix E. While the PRF keys are conditioned to be distinct, these steps add $e_{KD} + K\eta(1-\eta^{512})$; removing that conditioning costs δ_G, charged once. The external Module-LWE instance is never conditioned. Summing these costs gives (F.1). □

Appendix E gives the transport, hidden-family, and selective-answer derivations, together with their explicit interfaces to the implementation.

F.4 Extraction bound

Message-prehash collisions. Let Bad_μ be the event that two distinct complete message-hash inputs among the signing requests and the final forgery have the same 512-bit prehash, and let Δ_μ be its probability in the unconditioned ideal public-simulator experiment. A collision finder runs this complete experiment, records these classical encoded inputs and their prehashes, and outputs one colliding pair if one exists. The typed prehash collision finder gives $\Delta_\mu \le \min\{1, 400(Q_\mu+2)^3/2^{512}\}$, with $Q_\mu \le Q$. For sequential queries, the collision-output bound [6, Theorem 5.29] is

$$\frac{\left(2e\sqrt{10}(Q_\mu+1)^{3/2}+\sqrt{2}\right)^2}{2^{512}} < \frac{400(Q_\mu+1)^3}{2^{512}} \le \frac{400(Q_\mu+2)^3}{2^{512}}.$$

Here $2e\sqrt{10} + \sqrt{2} < 155/8 < 20$, using $e < 11/4$, $\sqrt{10} < 13/4$, and $\sqrt{2} < 3/2$. The cited bound requires a complete finder and a fresh oracle, which we now justify. We evaluate Bad_μ in the unconditioned ideal public-simulator experiment used before measure-and-reprogram. Its message family remains a uniform 512-bit random function: challenge programming does not alter that namespace, and the initial advice is independent of the fresh message family. The bound Q_μ is a fixed upper bound on every execution of the complete collision finder, including preprocessing, failed requests, adapter uncomputation, and any confirmations actually performed. It is not an expected count or a count on successful branches, and the theorem's query offset does not replace any of these actual calls. We subtract this event's probability from the forgery probability after the comparison of real and simulated signing by rejecting bad outputs; this neither conditions the experiment nor bounds the distance between complete quantum states. Outside this event, the log of every programmed point gives either a new typed point or two distinct accepted recorded (challenge, signature bytes) pairs.

The short vector. The Pretzel response reconstruction with the challenge given instead of hashed, which maps accepted signature bytes to response vectors, has a canonical inverse on its image, as shown in Appendix B.5. The resulting nonzero SIS difference obeys

$$\|d\|_2 < \beta = 6739, \qquad \kappa = 2^{964091/32768}.$$

Lemma B.1 gives the radius, and the certified secant table gives κ. The canonical lift of Appendix B.5 also transfers response measurement to accepted bytes, and in the SUF pair argument it rules out distinct encodings of the same response.

Finite uniform selection. The reduction guesses the attacked setup position and the measure-and-reprogram branch using finitely many fair bits. To sample among n choices, use up to $J \geq 1$ trials of $\lceil \log_2 n \rceil$ fair bits, and go to a common zero-output branch if all trials fail. Let $\lambda_n = 1 - (1 - n/2^{\lceil \log_2 n \rceil})^J$, with $\lambda_1 = 1$. Set $K_* = K/\lambda_K$, and use $\lambda_{\mathrm{MR}} = \lambda_{2q_B+1}$. The measure-and-reprogram branches are $(\lfloor j/2 \rfloor, j \bmod 2)$ for $j < 2q_B$, and the output branch is $(q_B, 0)$ for $j = 2q_B$. Writing $b = \lceil \log_2 n \rceil$ and $r = 1 - n/2^b$, each choice $v < n$ has probability $\sum_{i=1}^{J} r^{i-1} 2^{-b} = \lambda_n/n$. Exhaustion contributes zero success, so the reduction pays the multiplicative factor $\lambda_n \geq 1 - 2^{-J}$, and $K \leq K_* \leq K/(1 - 2^{-J})$. Keeping all Jb trial bits permits reversible selection and cleanup, including on exhaustion.

Off-diagonal strict extraction. Extraction needs accepted responses to two different challenges. Use the common-workspace folded projector family of Appendix C.2, retaining the unused raw bit. Its average success is exactly the raw-label average. Lemma C.1 gives the sharp bound $T_\perp \geq f_C(\gamma)$, where γ is the prover's average acceptance probability over the C challenges, $T_\perp$ is the probability that two independent uniform challenges differ and both responses accept, and $f_C(x) = x(x - 1/C)_+^2/(1 - 1/C)$. For the strict extractor, take the actual pulled-back projectors $P_c = U_c^\dagger \Pi_{\mathrm{accept},c} U_c$ on its common initialized workspace. The continuation predicate for measurement hiding is *both responses accept and $c \neq d$*. Changing the first response measurement costs the same measurement-hiding error once for this predicate, and there is no additional diagonal γ/C subtraction. This improves the lower bound without changing the scheme or the assumptions. The extraction dependence remains cubic, and the bound is positive for $\gamma > 1/C$.

Theorem 5.6 in Section 5.3 states the bound using these definitions. Its oracle-simulation error δ_{sim} is specified in Section F.2, and its circuit-precision errors $\varepsilon_{\mathrm{iso}}$, ε_ϕ, and δ_u in Section F.5.

Proof of Theorem 5.6. Before measure-and-reprogram [13], subtract the distance between complete outputs of real and simulated signing and the probability of the typed collision event. The potential target position is sampled independently and fixed before any challenge is answered. Check acceptance and eligibility before

the response measurement in which the hybrids differ, so that invalid outcomes and outcomes that are never reached have identical channels; sum over the signed partition into positions before taking absolute values. The strict extractor runs the prefix, obtains a response, runs the inverse, and obtains a second response [14,42]. The honest-key comparison of Appendix B.5 bounds its measurement loss for the off-diagonal predicate, yielding f_C above. No normalized retry overhead is added to the simulation, which already includes aborts. The hybrid that replaces the key relation gives the uniform normal-form SIS instance. Each key and auxiliary tester implants its instance from the start, with preprocessing included.

Only after this ideal-oracle argument, replace the oracles of each complete tester by finite ordinary oracle simulators. This costs δ_{sim} for a SIS output and $2\delta_{\mathrm{sim}}$ for an MLWE advantage, giving $3\delta_{\mathrm{sim}}$ for the pair term and $(7+2\kappa)\delta_{\mathrm{sim}}$ for one SIS term, three key-MLWE terms, and one weighted auxiliary-MLWE term. The precision comparison of Section F.5 adds the displayed 2κ contribution. This proves (45). □

F.5 Finite precision and interpretation

These precision parameters belong to the proof's algorithms, not to the signer.

For integer $B \geq 0$, use the cutoff D, count R and precision p of Appendix B.4; that construction gives $\varepsilon_{\mathrm{iso}} < 2^{-B}$ on accepted superpositions with arbitrary references, with the work bound stated there.

Compute the auxiliary modular inner product once into 13 bits; 13 logical rotations then give $\varepsilon_\phi \leq 13\,2^{-p_\phi}$. For the 1024 classical auxiliary coins, use b fair bits modulo 7681 per coordinate. With $r_b = 2^b \bmod 7681$,

$$\delta_u \leq \min\{1, 1024 r_b(7681 - r_b)/(7681\,2^b)\} \leq \min\{1, 1024 \cdot 7681/(4 \cdot 2^b)\}.$$

Taking $b = B + 23$ gives $\delta_u < 2^{-B-2}$. These are errors of classical channels; this construction has no additional rejection factor for phase sampling.

Corollary F.4 (Explicit bound on precision error)**.** Let $\zeta > 0$ and $\bar{\kappa} = 2^{30} > \kappa$. Choose integers $B \geq 0$ and $\lambda_c \geq 1$ satisfying

$$9\bar{\kappa}\,2^{-B} \leq \zeta, \qquad 2(7+2\bar{\kappa})2^{-\lambda_c} \leq \zeta.$$

Use the feature parameters above, $p_\phi = B + 4$, $b = B + 23$, and the matrix-coefficient trial count specified for λ_c in Section F.2. Then the precision contribution inside the brackets of (45) is at most ζ.

Proof. The matrix oracle simulator gives $\delta_{\mathrm{sim}} \leq 2^{-\lambda_c}$, or $\delta_{\mathrm{sim}} = 0$ when it is unused. The other bounds give $\varepsilon_{\mathrm{iso}} < 2^{-B}$, $\varepsilon_\phi \leq 13\,2^{-B-4} < 2^{-B}$, and $\delta_u < 2^{-B-2}$. Consequently, by the two conditions on B and λ_c, their total contribution is at most

$$(7+2\bar{\kappa})2^{-\lambda_c} + \tfrac{9}{2}\bar{\kappa}\,2^{-B} \leq \zeta.$$

□

In the theorem, this total contribution is multiplied by K_*. The error in comparing real and simulated signing, the SUF pair term, and the work and storage of the complete algorithms must still be charged separately.

F.6 Sharper bounds for extraction and ordinary unforgeability

Extraction using the filter of Lemma C.2. Put $\kappa_0 = \sqrt{\kappa} = 2^{964091/65536}$. The filter of Lemma C.2 bounds the extraction event directly, rather than every effect of measuring the response. Let e_R, e_U bound the errors in output probability of the finite phase and filter circuits, relative to the ideal phase and filter, with a real and with a uniform auxiliary sample. The tester costs include table construction, filter evaluation, finite phase synthesis, and all retained registers.

Corollary F.5 (Extraction bound using the filter)**.** Keep the experiment, distance between real and simulated signing Δ_F, collision loss Δ_μ, K_*, q_B, λ_{MR}, oracle-simulation error δ_{sim}, and pair term P_{pair} of Theorem 5.6. In particular, $P_{\mathrm{pair}} = 0$ for EUF, and $P_{\mathrm{pair}} = K_*(s_{\mathrm{pair}} + a_{\mathrm{key,pair}} + 3\delta_{\mathrm{sim}})$ for SUF. Let $a_{\mathrm{key},0,+}$ be the key-MLWE advantage of the sequential extraction tester *without* the response measurement. Let $s_{\mathrm{ext},+}$ be the success probability of the sequential extractor, with the first response measured, run directly on a uniform relation matrix, and let $a_{\mathrm{aux},+}$ be the advantage of the tester using that filter. These are specific complete algorithms, whose costs may differ from those of the testers in Theorem 5.6. Put $\epsilon_0 = [\epsilon - \Delta_F - \Delta_\mu - P_{\mathrm{pair}}]_+$. Then

$$f_{2^{255}}\left(\frac{\lambda_{\mathrm{MR}}\epsilon_0}{(2q_B+1)^2}\right) \le K_*\big[a_{\mathrm{key},0,+} + \kappa_0(s_{\mathrm{ext},+} + a_{\mathrm{aux},+} + e_R + e_U) + (2+3\kappa_0)\delta_{\mathrm{sim}}\big]. \tag{F.2}$$

The right-hand side of (45), with its own tester costs and precision, remains an alternative, and one may take the smaller of the two. The implementation assumptions of the theorem still apply.

The proof is in Appendix C.2. The coefficient κ_0 multiplies both the auxiliary Module-LWE advantage and the extracted Module-SIS success probability, but not the key Module-LWE advantage. It reduces the logarithm of the coefficient from about 29.42 to 14.71 bits. This is not an equal improvement in the forgery bound, which depends on the cube root of the right-hand side.

Ordinary unforgeability. For a forgery on an unsigned message, a return test needs only one response instead of two. Without measurement, the test succeeds with probability at least the square of the prover's success; measuring the response confines the challenge to the two raw values with the same fold. The following additional assumption makes the return experiment well defined.

Let R be a register of 256 qubits with basis $\{|r\rangle : r \in \{0,1\}^{256}\}$, put $N = 2^{256}$ and $Q = |+_N\rangle\langle+_N|_{\mathsf{R}} \otimes I$ for the uniform superposition $|+_N\rangle$. For a fixed classical outcome of the measure-and-reprogram step, let U_r be the circuit run after the challenge when the measured point is programmed to the raw value r, and let Π_r be the acceptance projector of Appendix B.5 for r, including the check that a matching key handle was returned. As in Theorem 5.6, the collision event Bad_μ is handled separately, outside Π_r.

Coherent challenge interface. For every supplied r, the computation after the challenge is one fixed circuit U_r. It leaves no garbage except the answer registers kept by verification requests in Section F.2, which only $U_r^\dagger$ removes. Its first gate copies R, by CNOT gates, into the value field of the saved input–value pair for the programmed challenge, and no other gate writes to R. The implementation must satisfy the correspondence of Section F.2 for this circuit, with the value field treated as part of its input. The correspondence then extends linearly to superpositions of r. This assumption is in addition to those of Theorem 5.6.

Corollary F.6 (Quadratic bound for ordinary unforgeability)**.** Keep the experiment, Δ_F, Δ_μ, K_*, q_B, λ_{MR}, δ_{sim}, and the filter errors e_R, e_U of Corollary F.5, and assume the coherent challenge interface. Let ϵ be the EUF forgery probability, put $\epsilon_0 = [\epsilon - \Delta_F - \Delta_\mu]_+$ and $g = \lambda_{\mathrm{MR}}\epsilon_0/(2q_B+1)^2$. Let $a_{\mathrm{key},R}$ and $a_{\mathrm{aux},R}$ be the advantages of the key-MLWE and auxiliary-MLWE return testers constructed in the proof, at their own complete costs, and put

$$B_R = K_*\big[a_{\mathrm{key},R} + \kappa_0\big(a_{\mathrm{aux},R} + e_R + e_U\big) + (2+2\kappa_0)\delta_{\mathrm{sim}}\big].$$

Then

$$g^2 \le 2^{-240} g + B_R. \tag{F.3}$$

In particular $\epsilon \le \min\{1, \Delta_F + \Delta_\mu + (2q_B+1)^2\lambda_{\mathrm{MR}}^{-1}(2^{-240} + B_R^{1/2})\}$.

The proof and complete query accounting are in Appendix C.2. The corollary concerns EUF only; for strong unforgeability the pair term P_{pair}, including its Module-SIS success term, remains. The coherent challenge interface is an additional requirement on the implementation. It does not follow from equality of the completed outputs in the comparison of real and simulated signing alone.

G Further Security Estimates

The attack estimates use the secret distribution before key generation rejects candidates that violate the key-norm restriction (3). We first show that conditioning on this restriction increases the success probability of any fixed attack by at most the factor $1/(1-\eta_{\mathrm{shell}})$, whose base-two logarithm is below 2^{-190}, then give the attack models and their costs.

Write KeyGen_0 for key generation with one unrestricted CBD_1 candidate, $\mathsf{KeyGen}_{\mathrm{shell}}$ for key generation conditioned on (3), and η_{shell} for the probability that a KeyGen_0 candidate violates (3).

Lemma G.1 (Search attacks transfer to the conditioned key distribution)**.** Let W be any event determined by the key pair and the coins of a fixed algorithm, such as recovering the key within a fixed time bound. For the cap (3),

$$\Pr_{\mathsf{KeyGen}_{\mathrm{shell}}}[W] \le \frac{\Pr_{\mathsf{KeyGen}_0}[W]}{1-\eta_{\mathrm{shell}}}.$$

The corresponding logarithmic loss is below 2^{-190} bits.

Proof. Let S be the event that the candidate satisfies both inequalities in (3). Then $\Pr_0[W] \ge \Pr_0[W \cap S] = (1-\eta_{\mathrm{shell}})\Pr_{\mathrm{shell}}[W]$. To bound η_{shell}, note that squared CBD_1 coordinates are independent Bernoulli$(1/2)$ variables, so $\|s_0\|_2^2$ and $\|e\|_2^2$ are independent Binomial$(1024, 1/2)$ variables. Sum the exact integer weights $\binom{1024}{x}\binom{1024}{y}$ over $0 \le x, y \le 1024$ satisfying $120{,}000 \le 23{,}716 + 196x + 121y \le 244{,}553$. Subtracting this sum from 2^{2048} and dividing by 2^{2048} gives $\eta_{\mathrm{shell}} < 2^{-192}$, with no truncation or rounding. For the logarithmic loss, use $-\log_2(1-\eta) \le \eta/((1-\eta)\ln 2)$ and $\ln 2 > 1/2$. □

Returned keys are also conditioned on a successful dither search. In the ideal independent-stream model of Appendix D.2 that search fails with probability d_4^{2048}, where $d_4 < 2^{-742}$, so this conditioning multiplies the bound by at most $1/(1-d_4^{2048})$.

The lemma bounds the success of any fixed key-recovery algorithm. For a distinguisher that compares keys with a uniform reference distribution, the triangle inequality instead gives an additive loss of at most η_{shell}. Each key replacement in the reduction pays this loss, and the finite reduction determines how many replacements there are.

Estimator inputs and attack models. The attack portfolio behind Table 5 covers primal uSVP, BDD and its hybrids, coded BKW, dual and dual-hybrid attacks, and the lattice SIS attack. Each instance is optimized under MATZOV gate counts and under classical ADPS16 (0.292β). The table reports the lowest cost returned for each model. These are heuristic estimates from the Lattice Estimator, not security lower bounds.

For ML-DSA, the key instances use the unrounded public-key equation and uniform secret and error coefficients in $[-\eta, \eta]$. This models key recovery by an attacker who has seen signatures: the published key is rounded, but FIPS 204 explains how a small number of signatures reveals the omitted low bits [35, Section 6.1]. The table estimates the lattice attack that follows. Its ML-DSA SIS rows use the infinity-norm instances for strong unforgeability.

For Pretzel, the inputs are the unrestricted CBD_1 key instance, the auxiliary Module-LWE instance used in the measurement proof, and normal-form Module-SIS. For Pretzel-(3,3), the table's SIS radius 6913.835 follows from the separate Euclidean caps. For Pretzel-(5,5), the squared radius is 58,848,400 (Section I.3). For Pretzel-(4,4), the table uses the sharper radius 6739 of Theorem 5.1, which combines these caps with the secant check; at this radius the estimator, at the same revision and with Euclidean SIS parameters, selects block size 950. The caps alone give radius 6913.835 for Pretzel-(4,4) as well. The key and auxiliary instances are unchanged.

H Pretzel-(3,3) Parameters and Conditional Security Bounds

H.1 Parameters and measured costs

Signatures occupy 1160 bytes. The public-key body occupies 1312 bytes; the self-identifying format adds a 14-byte parameter-set identifier. A protocol that selects this parameter set externally can use the 1312-byte format, and the API for the fixed parameter set then reconstructs the tag internally before hashing, so the specified hash inputs are unchanged. Size comparisons must state which format they use.

The challenge consists of four blocks of length 64, each of weight 18. Each block is determined by 51 hash bits: read them as a little-endian rank in $[0, 2^{51})$ and take the weight-18 binary string with that rank, counting from zero, in lexicographic order with $0 < 1$. Since $\binom{64}{18} > 2^{51}$, this map is injective, giving 2^{204} equally likely challenges of total weight 72. The signer therefore performs 72 transport steps. For this parameter set, the attempt specification specializes the transport procedure of Section 4.3 and Appendix D to exactly 72 active three-outcome transport steps. Since every challenge has weight 72, no dummy Bernoulli tests are appended; the 128-test padding of the Pretzel-(4,4) attempt does not apply here. Before transport, the implementation prepares 80 integer logarithms. Each approximates the logarithm of a random number, with ambiguous conversions recomputed using longer arithmetic, as in Appendix D.4. The 72 transport steps use 72 of these integer logarithms. Computing any of the 80 integer logarithms can cause an abort; the assumed numerical bound includes these failures.

Pretzel-(3,3) uses a simpler encoding than the one in Section 4.5. The leading coefficients are signed six-bit fields, occupying 192 bytes, and the 768 nonleading head coefficients are signed low bytes. A bounded list of exceptional positions marks where to add 256 to a negative low byte or subtract 256 from a positive one; an exceptional zero is rejected. The 768 local labels use the same list of four candidate labels as Section 4.5, with low index bits and a list of exception positions. The encoding allows sixteen head exceptions and forty local-label exceptions, whose fields occupy $1 + 16 \cdot 10/8 = 21$ and $1 + 40 \cdot 10/8 = 51$ bytes, respectively. The signature length is therefore $32 + (192 + 768 + 21) + (96 + 51) = 1160$ bytes. The leading and nonleading coefficient caps are 31 and 255, and the local residual cap is 450. The ordering and zero-padding checks are unchanged.

Parameters. The ring degree and modulus remain $n = 256$ and $q = 7681$. Secret and error coefficients are sampled from CBD_1 before the key-norm restriction is applied. As for the other parameter sets, the key-norm statistic uses weights (23,716, 196, 121), while the weighted squared norm $E(s)$ uses (9216, 196, 121), so the two intervals below differ by 14,500.

Quantity	Pretzel-(3,3)
Module dimensions (ℓ, m)	$(3, 3)$
Quantizer levels	34
Key-norm statistic	[94,843, 196,045]
Weighted squared norm $E(s)$	[80,343, 181,545]
Transport envelope M	65,544/65,536
Euclidean head/residual caps	(1593, 3068)
Challenge cardinality	2^{204}
Transport steps / prepared integer logarithms	72/80
Leading range-draw cap	352 words
Signing attempt cap	512

The base of the target distribution is the dyadic rational $\theta_{33} = T_{33}/2^{256}$, with hexadecimal numerator

$$T_{33} = \texttt{0xffffebe71e0d482af9a42878e0fbcb58}\\ \texttt{f7663bf7f0afda571e6a48849f6572a5}.$$

This is the same base as for Pretzel-(4,4). The envelope $\mathcal{M}(b)$ increases with b, and the smaller key-norm interval above bounds $E(s)$ by 181,545 rather than 230,053, so the smaller constant $M = 65{,}544/65{,}536$ suffices: $\mathcal{M}(181{,}545) \approx 1.0001117$. The integer secant tables are inherited from Pretzel-(4,4), with threshold 30,850,912. The separate Euclidean caps give the conservative bound $\beta^2 = 4(1593^2 + 3068^2) = 47{,}801{,}092 < 7681^2$ on the difference of two responses, under the same extraction interface as Pretzel-(4,4). That interface remains an assumption; Table 5 gives the estimated hardness of the resulting SIS instance. Pretzel-(3,3) uses this separate-cap bound; the tighter radius that Pretzel-(4,4) obtains from the secant check is not used here.

Dither selection for Pretzel-(3,3). After fixing the core key, draw a 32-byte candidate-start seed; failure of this entropy request returns an error. Use the candidate order and bounded dither expansion of Appendix D.2, specialized to three residual polynomials: an expansion reads at most 1536 words and succeeds upon obtaining 768 values. Return the first candidate whose expansion succeeds, or fail if all 2048 candidates fail, without replacing the core key. No public sample responses are drawn or tested. Serialize the selected seed with the core key; public-key preparation expands it.

Reference sample for the row bound. Key generation draws no sample responses. The following reference sample specifies the row-quality statement of Lemma H.1, which is used for availability. Fix a core key, a dither δ and a challenge c, and draw 1792 independent uniform 32-bit words: 256 for the leading coefficients, 768 for the other head coefficients and 768 for the errors.

Conditional on challenge c, the ideal row has product law D_c with base θ_{33} and weights (9216, 196, 121): leading coordinates have parities c, and the other coordinates are unrestricted. For each parity or weight let p_z be the full normalized ideal Gaussian probability of integer z. Allocate $n_z = \lfloor 2^{32} p_z \rfloor$ words to z, omitting zero entries, and store $e_z = \sum_{u \le z} n_u$ in increasing order of z. A word $w \in [0, 2^{32})$ selects the first endpoint $e_z > w$. If there is none, the sample fails without resampling. These are sums of individually rounded-down masses, not rounded cumulative probabilities. Every finite nonfailure row mass is at most its ideal product mass; summing over any public predicate preserves this domination without renormalization.

Write the sampled head and error as $y_h = (y_0, y_s)$ and y_e. For the fixed core key and dither compute

$$x = A_h y_h + \delta, \qquad U = Q_{q,34}(x + y_e), \qquad r = \mathrm{center}_q(C(U) - x).$$

A sample succeeds exactly when all coordinates exist, the leading, other head and residual caps $(31, 255, 450)$ hold, the head and residual Euclidean norms are at most $(1593, 3068)$, and the integer secant sum is at most 30,850,912. It must also pass the complete canonical encoding checks above: every label is recoverable, at most sixteen head exceptions and forty local-label exceptions occur, and the ordering, exceptional-zero and padding rules hold. No Fiat–Shamir hash fixed point is required for the challenge.

Lemma H.1 (Rank-three spread and row quality)**.** Let $q = 7681$ and $A_0 \in (\mathbb{F}_q[X]/(X^{256}+1))^{3\times 3}$. Define $a_{i,k}$ and $|v|_q$ as in Lemma D.3, for the 768 output coordinates i and the 768 input coordinates k of the head outside its leading polynomial. The condition $G_3(A_0)$ is

$$\#\{k : |t a_{i,k} + s a_{j,k}|_q = u\} \le 54 \quad (i \ne j,\ (t,s) \in \mathbb{F}_q^2 \setminus \{(0,0)\},\ 0 \le u \le 3840).$$

Let F_3 be the success event of one reference sample above. For every A_0 satisfying G_3, every $b \in R_q^3$, every fixed $\delta \in R_q^3$ and every fixed $\beta \in \{0,1\}^{256}$,

$$\Pr[\neg F_3 \mid A_0, b, \delta, \beta] < \varepsilon_3 := 16575/10^6.$$

The probability is over fresh independent sample words. The dither and parity are fixed before these words; this is a pointwise statement, not a statement about choosing the dither after observing a sample. The ideal product law D_c has at least the same successful mass. For 2304 independent uniform matrix coefficients,

$$\Pr[\neg G_3] \le \varepsilon_{G,3} := \frac{(q+1)^2}{2}\left(768\binom{384}{28}(2/q)^{28} + 768\binom{768}{55}(2/q)^{55}\right) < 2.701 \cdot 10^{-48}.$$

This exceptional probability concerns an honestly expanded matrix in the ideal uniform-matrix model, and key generation does not test G_3. The vector b may depend on A_0, including $b = A_0 s_0 + e$.

Proof. Follow the proof of Lemma D.3 with three residual polynomials. Condition on all coordinate samples existing and fix the leading block. Condition G_3 permits at most 54 factors in any absolute-frequency bin. Among 767 factors the largest possible product of the sorted upper bounds on the SECRET Fourier magnitudes is at most $\nu_3 < 4.442 \cdot 10^{-15}$ by the exact certificate, and this covers fixing any one SECRET coordinate. Fourier inversion gives

$$d_{\mathrm{TV}}((x_i, x_j), U_q^{\otimes 2}) \le \tau_3 := (q^2-1)\nu_3/2 < 1.311 \cdot 10^{-7},$$

with the same marginal and conditional consequences as at rank four. No independence of all residual coordinates is asserted.

For a capped residual statistic $Y \in [0, B]$ with reference mean and variance μ_Y, v_Y under a uniform context and normalized ERROR draw, the moment bounds become

$$M_Y := 768(\mu_Y + B\tau_3), \qquad V_Y := 768 v_Y + 3 \cdot 768^2 B^2 \tau_3,$$

and the combined score adds the head means and variances and $4 \cdot 1024 \cdot 768 H_{\max} B \tau_3$ to the variance. The exact moments give

	M upper bound	V upper bound	$T+1$
residual norm	5915749.390	83972187943.502	9412625
combined score	22016058.562	675183301005.394	30850913

and Cantelli's inequality bounds both tails as before. The head norm uses the Chernoff bound of that proof with 256 leading and 768 SECRET coordinates, $T = 2537649$ and $b_0 = 10000800/10000000$.

The encoding limits use binomial tails. Each SECRET coordinate lies outside $[-128, 127]$ with probability below 0.005544, independently, so more than sixteen head exceptions occur with probability below $2.330 \cdot 10^{-6}$. Conditional on the whole head every residual context is fixed, and each local-label exception depends on its own ERROR word; over every context its probability is below 0.0277169. The number of such exceptions is therefore stochastically dominated by a binomial variable with 768 trials, and more than forty occur with probability below $7.372 \cdot 10^{-5}$. This bound is uniform in the head, so it holds without the conditioning.

The missing-word union bound is 969/8388608: there are 24 missing words per leading coordinate, 293 per SECRET coordinate and 345 per ERROR coordinate. Cap-tail words number 4153344 over the leading coordinates, 76800 over the SECRET coordinates and at most 9216 over the residuals in any context, out of 2^{32}. Enumeration of every context and every ERROR atom gives no unrecoverable label within residual cap 450. The six upper bounds, rounded upward, are

missing samples	0.000115513802
coordinate caps and recovery	0.000987052918
head norm	$5.197 \cdot 10^{-19}$
residual norm	0.006820291536
combined score	0.008575952056
both encoding capacities	0.000076048071

Their sum is below $0.016574858384 < \varepsilon_3$. Pointwise domination of the finite atoms by the ideal Gaussian masses proves the ideal-product assertion without normalizing finite successful mass.

For the matrix statement, the reduction of that proof leaves $(r, 0), (r, d)$ for $1 \le d \le 128$ (384 representatives) and $(r, 0), (s, d)$ for $r < s$, $0 \le d < 256$ (768 representatives). A cross-polynomial representative has 768 independent uniform projections, and a union over 55-element subsets gives $\binom{768}{55}(2/q)^{55}$. For a same-polynomial representative each of the two matchings has 384 projections of disjoint coefficient pairs; more than 54 values in a bin put at least 28 in one matching, which gives $2\binom{384}{28}(2/q)^{28}$. The union over representatives, $q + 1$ directions and $(q + 1)/2$ bins is $\varepsilon_{G,3}$, evaluated in exact rational arithmetic. An accompanying checker and certificate verify these computations. □

The bound of Lemma H.1 does not extend to every matrix. For $A_0 = 0$, $b = 0$ and $\delta = 0$ every context is zero and every center has magnitude at least 112, so the residual squared norm is at least $768 \cdot 112^2 = 9{,}633{,}792 > 3068^2$ and every sample fails.

Measured costs. Table 8 compares Pretzel-(3,3) with AWS-LC's ML-DSA-44 on macOS (Apple M2 Pro), with 32-byte messages. The table reports mean thread CPU time. Verification including public-key preparation includes building and releasing the prepared public key within the measured operation. Both implementations use an empty context. ML-DSA-44 signatures occupy 2420 bytes.

Pretzel signs 9% faster than ML-DSA-44, verifies 13% faster with an unpacked key and 28% faster including key preparation. Key generation takes about 0.028 ms, about 1.5 times the 0.019 ms of ML-DSA-44. Pretzel-(3,3) thus has smaller signatures than ML-DSA-44 and faster signing and verification.

Changes to the security analysis. The smaller challenge space changes the extraction function to $f_{2^{204}}(x) = x(x - 2^{-204})_+^2/(1 - 2^{-204})$. The secant threshold gives the coefficient of Lemma C.2 $\log_2 \kappa_0 = 964{,}091/65{,}536$, as for Pretzel-(4,4). This coefficient multiplies the auxiliary Module-LWE and Module-SIS terms of the reduction; it is not subtracted from the estimated key-recovery cost.

Operation	Pretzel-(3,3)	ML-DSA-44	Time ratio (interval)
Sign (prepared secret key)	38.537	42.496	0.9075 [0.8989, 0.9154]
Verify with preparation	12.804	17.793	0.7196 [0.7156, 0.7235]
Prepare public key	8.871	13.494	0.6571 [0.6523, 0.6627]
Verify prepared	3.827	4.396	0.8663 [0.8500, 0.8850]
Generate key	28.26	19.28	1.4660 [1.4595, 1.4726]

Table 8: Pretzel-(3,3) and AWS-LC ML-DSA-44 mean thread CPU times in microseconds. Brackets give 95% intervals. Verify prepared checks a signature with a public key that has already been prepared, that is, unpacked into the matrix and offset used in verification; Prepare public key is that unpacking; Verify with preparation does both, starting from the stored key.

Let D_{attempt} be the statistical distance between one finite signing attempt and its ideal counterpart, with aborts counted as outputs. Assume ideal private random tapes, correct primitives, and the transport interface for all outputs and retained state described below. We take $D_{\text{attempt}} < 2^{-169}$ as the Pretzel-(3,3) assumed numerical bound of Assumption F.3, including the shared-tail comparison. The interface for all outputs and retained state requires each later step to use the decisions already made and the current state, without retesting earlier coins at changed correlations. The observations are the returned signature or abort and the state kept for later calls; temporary lists of pending values, discarded coins and memory-access traces are excluded. Reset prevents later calls from observing private state discarded by a failed attempt.

The bound includes aborts, overflow of the mask sampler's lists of pending nonleading and leading coefficients, finite-precision transport effects, and the cap on words drawn by the leading-coefficient sampler. At most S signing calls make at most $512S$ attempts, which contribute at most $512SD_{\text{attempt}}$. This is one term of the forgery bound, not a security level. Key generation, oracle replacement, extraction, and the lattice assumptions enter the complete bound separately (Section H.2).

Deferred step decisions. The interval and correction decisions follow the mechanism in Appendix D.4, with the Pretzel-(3,3) parameters. Integer interval membership determines one branch. A short (coarse) polynomial evaluates the correction comparison; an active step is *marked* when this comparison lies in its uncertainty band. Such a step is applied provisionally and checked after all steps with a longer (fine) polynomial. For each marked step, the implementation saves the inner product $\langle z, X^j s\rangle_W$ before the update (the step's *correlation*), the integer logarithm, and all six original coin words. The final check uses only these saved values and never substitutes a correlation computed later.

For a marked step, let P be the add test (membership of the integer logarithm in the union of intervals) and F the subtract test computed with the fine polynomial, as in Section I.2. The provisional decision is plus if P and minus otherwise, with the specified sign swap. The final check rejects exactly when $\neg P \wedge \neg F$. Thus every provisional decision that survives the final check equals the decision the fine polynomial would have made immediately. For unmarked steps, the certified error bound of the coarse polynomial ensures that its decision agrees with the fine one.

Set aside failures in computing the integer logarithms. Induction over the steps, up to the first rejection, then couples deferred checking with immediate checking on the same coins. If no step rejects, both make the same updates and release the same signature bytes; if a step rejects, both attempts abort. The number of marked steps saved for later checking is bounded; exceeding this limit can cause an additional abort, charged by the bound on that event. The coupling compares the released value or abort and the state kept after reset; it does not claim that discarded private states agree after a rejection.

The statistical replacements are made in a fixed order. First, charge failures in computing the integer logarithms and exceeding the limit on marked steps saved for checking on the finite proposal support,

including the eight unused integer logarithms. Second, in the reference that checks each step immediately, replace the step coins, which share a common tail, by independent words. Third, compare with ideal transport. Finally, replace the finite proposal by the unbounded parity-restricted Gaussian; applying the same total mathematical map to both cannot increase their distance.

H.2 Conditional security specialization

Pretzel-(3,3) query allocation. Let $1 \le K \le 2^{64}$ bound the number of key-generation calls, and use the request and query bounds S, V, h, r and $A_{\mathrm{att}} = 512S$ of Section F.1. The budgets below allow each key-generation call 7448 oracle calls including uncomputation. Dither selection as specified in Section H.1 uses at most 2572 encoded hash inputs, or 5144 calls: 512 key-seed streams, 2048 dither streams and 12 inputs for matrix, integrity and selected-dither work. The allowance is therefore conservative.

Use the same conservative query budgets as for Pretzel-(5,5):

$$\begin{aligned} R &= r + 7448K + 32A_{\mathrm{att}} + 16V + 8S + 32, \\ D_M &= 12K + 8(V + S + 1), \\ Q_{\mathrm{bin}} &= 180R + 6h + 8 + 6A_{\mathrm{att}}, \\ Q_{\mathrm{mat}} &= 96R + 6D_M, \qquad Q = Q_{\mathrm{bin}} + Q_{\mathrm{mat}}. \end{aligned}$$

These budgets include uncomputation and exceed the per-attempt, verification, and message-hash allowances of Section F.2. They use the same multipliers for replacing oracles by oracle simulators and the same allowance for three executions of each constructed algorithm. Pretzel-(3,3) does not verify a signature before releasing it; even so, the bound $q_B = 2h+2r+2V+2S+2$ conservatively allows one verification per successful signing request. The requirements of that section on the complete retained output and on immutable keys also apply.

Pretzel-(3,3) matrix oracle simulator. Matrix expansion uses two output streams, which supply 1280 and 1024 accepted values, respectively. For stream $\ell \in \{0, 1\}$ and accepted-value index k, set $p = 2\lfloor k/256 \rfloor + \ell$ and place the value at matrix row $\lfloor p/3 \rfloor$, column p mod 3, and spectral coordinate k mod 256. This is a bijection onto the 2304 matrix coordinates. The construction of Section F.2 that answers matrix and raw-stream queries consistently applies with these required numbers of accepted values in each stream. Its argument, conditioned on the pattern of rejected values, preserves the joint law of the matrix and the raw stream, including queries made before key generation.

The oracle simulator works over the same degree-20 extension field as for Pretzel-(4,4) and uses $\lceil 2304/20 \rceil = 116$ random polynomials, retaining 2304 output coordinates and discarding 16 padding coordinates. Each polynomial has degree $2Q_{\mathrm{mat}} - 1$, and hence $2Q_{\mathrm{mat}}$ extension-field coefficients, for $4640Q_{\mathrm{mat}}$ base-field coefficients in total. For $Q_{\mathrm{mat}} > 0$, sampling each coefficient by bounded rejection, as for Pretzel-(4,4), contributes

$$\delta_M = \min\{1, 4640Q_{\mathrm{mat}}(511/8192)^{t_c}\}.$$

When its query budget is zero, this oracle simulator is omitted. The rejection cap t_c applies to the proof's coefficient sampler, not to the scheme's matrix expansion. The experiment observes the returned matrix and the state kept for later calls, not the running time of the implementation's rejection sampling.

Raw and typed challenge access. The challenge map reads the low 204 bits of a 256-bit hash output, in little-endian bit order, as four consecutive 51-bit ranks. The other 52 bits do not affect the challenge. The simulation must answer raw hash queries and typed challenge queries consistently. To preserve their joint law, the typed oracle supplies a uniform 204-bit rank, and independent auxiliary bits fill the unused suffix.

At a challenge hash input, the raw answer has the rank in bits 0 through 203 and auxiliary bits in positions 204 onward. In byte 25, the low nibble belongs to the rank and the high nibble to the auxiliary suffix. Prefix queries return the requested bits of this same persistent answer.

Since the four block maps are injective, the typed challenge is uniform on its 2^{204}-element image. The rank followed by the independent suffix is a uniform raw string whose image under the challenge map is the typed challenge. Repeated queries return the same rank and suffix. When the simulation programs a challenge, it sets only the 204 rank bits; the other bits keep their auxiliary values. Programming all 256 raw bits with a zero-padded challenge would break this consistency. The simulator is the clean compute, XOR, and uncompute construction of Section F.2, so its equality on basis states extends to superposition queries, under the same conditions on the oracle simulators.

Pretzel-(3,3) public response sampler. The signing simulation uses the coordinate construction of Section I.2 on 1792 coordinates, with $\theta = \theta_{33}$ and weights $(9216, 196, 121)$. Its independent acceptance bit has probability

$$\alpha_{33} = \lambda_{33}^{72}, \qquad \lambda_{33} = \frac{Z_{o,33}}{M_{33} Z_{e,33}}, \qquad M_{33} = \frac{65{,}544}{65{,}536},$$

where $Z_{e,33} = \sum_{x \in 2\mathbb{Z}} \theta_{33}^{9216x^2}$ and $Z_{o,33} = \sum_{x \in 2\mathbb{Z}+1} \theta_{33}^{9216x^2}$. The exponent is the challenge weight: the Pretzel-(3,3) attempt specification in Section H.1 makes exactly 72 transport steps, with no dummy tests. The eight unused integer logarithms contribute only to the finite signer's abort comparison, not to this ideal gate.

Here is the complete public-sampler error budget. The selected base satisfies $121(1 - \theta_{33}) > 1/9216$ and $\theta_{33} \geq 1 - 2^{-19}$. The tail and directed-normalization arguments of Section I.2 therefore apply at the same 640 fractional bits and truncation radius 4096. There are fewer than $1792 \cdot 8192 < 2^{24}$ internal CDF boundaries. Rounding their lower endpoints to 352 bits costs less than $2^{-328} + 2^{-556}$ in total variation; the joint omitted tail is below 2^{-2387}. The gate ratio has an enclosing interval of width $w < 2^{-567}$, clipped to $[0, 1]$. Compute its 72nd power as $x^{64}x^8$, retaining the third squaring's result, using six squarings and one multiplication. With $u = 2^{-640}$, outward interval multiplication adds at most $2u$ to the sum of input widths, so the final width is at most $72w + 142u < 2^{-560}$. In exact arithmetic the same sensitivity follows from $|x^{72} - y^{72}| \leq 72|x - y|$ on $[0, 1]$. Rounding the lower gate threshold to 352 bits and using an independent uniform 352-bit word adds less than $2^{-352} + 2^{-560}$. Consequently the error used below is

$$e_{\mathrm{pub}} = e_{\mathrm{pub},33} < 2^{-328} + 2^{-556} + 2^{-2387} + 2^{-352} + 2^{-560} < 2^{-326}.$$

This compares the complete coordinate-and-gate distributions before the common map to signature bytes, filters and reset, with failures retained. It does not condition on signing success. The signer's mask-sampling error is part of D_{attempt}. Key generation draws no public sample rows.

For these parameters, directed integer intervals give $0.991249980 < \alpha_{33} < 0.991249981$.

Public setup in the rank-three comparisons. For every canonical core key, including an external uniform column, Section H.1 defines a total deterministic map from that core, the candidate-start seed and the dither oracle to the first candidate whose bounded expansion succeeds, or failure. In the rank-three specialization of Assumption F.1, item 4 requires this candidate order, bounded expansion and stopping rule. Each subroutine is a deterministic function of its inputs with the stated cleanup and exclusive-storage contracts. Induction over the candidates therefore preserves selected dithers, errors and entropy order in both experiments.

The dither inputs have their own domain prefix, so they are outside the challenge domain. The fixed-table induction of Lemma F.2 applies to these inputs: challenge programming does not change the selected dither. The private PRF key is drawn independently of the core and the public setup seed; dither selection reads neither it nor private signing streams.

Both sides of every rank-three comparison run this same dither selection, keeping the selected dither and failure outcome. Once signing is public, the common continuation is witness-free on every canonical core key. Applying it to the restricted and unrestricted core laws contracts their distance and gives $K\eta(1-\eta^{512})$, including the common all-failed branch. Dither selection adds no error term and introduces no conditioning on setup success, on the spread condition of Lemma H.1, on row quality or on signing success. The same public map is used by the key testers and the predrawn-setup and extraction arguments.

Replacing signing by simulation for Pretzel-(3,3). The comparison of real and simulated signing bounds how much the adversary's view changes when signing is replaced by public simulation. Use the security experiment of Section F.1, and assume the interfaces of Appendix E, specialized to Pretzel-(3,3), for functional behavior, private labels, replay, reset, and unread rows. Each successful entropy request returns fresh uniform bytes, and entropy failures behave the same way in both experiments, independently of the secret. Dither selection and key replacement use the rank-three public continuation of Paragraph H.2. The finite sampler also requires the ordered comparisons of Section H.1 for computing the integer logarithms, deferred decisions, and shared-tail coins.

Let K, S, V, h, r and the oracle-simulation budgets be as above. Put $A_{\mathrm{att}} = 512S$, and let $N \le A_{\mathrm{att}}$ bound the number of distinct attempt names; an attempt's name is its complete input to the private randomness streams. Repeated names reuse their cached results, including rejections. Define $d(u) = 1 - (2^{256})_u/2^{256u}$ for $u \le 2^{256}$ and $d(u) = 1$ otherwise, with $d(0) = 0$. Set

$$\begin{aligned} \delta_G &= d(K), & e_P &= \min\{1, 4KQ^2/2^{256}\},\\ e_{KD} &= d(512K) + (1 - d(512K))\min\{1, 2048KQ^2/2^{256}\},\\ e_C &= \min\{1, 6NQ/2^{128} + \tbinom{N}{2}/2^{256}\}. \end{aligned}$$

Let η be the probability that one ideal unrestricted key candidate violates the key-norm restriction. The exact binomial count of Lemma G.1, with 768 secret and 768 error coefficients and the interval above, gives $\eta \approx 2^{-192.018} < 2^{-192}$. Use the per-attempt assumed numerical bound $D_{\mathrm{attempt}} < 2^{-169}$ of Assumption F.3, with the shared-tail conditions stated in Section H.1. The public response sampler, built with directed rounding, gives $e_{\mathrm{pub}} < 2^{-326}$. Set $\tau = 0$: the implementation's leading-coefficient sampler already stops after 352 words, and its exhaustion probability is included in D_{attempt}. Its entire stream is at most 2960 bytes, and the six private streams of an attempt together consume at most 8416 bytes, within the oracle simulator's output width. The comparison of real and simulated signing then gives

$$\begin{aligned} E_F &= e_P + e_{KD} + e_C + A_{\mathrm{att}}(D_{\mathrm{attempt}} + e_{\mathrm{pub}} + \tau) + K\eta(1-\eta^{512}),\\ \Delta_F &= \min\{1, \delta_G + (1-\delta_G)\min\{1, E_F\}\}. \end{aligned}$$

This comparison keeps the state retained for later calls and cached results of repeated names. Both sides run the same rank-three dither selection, with its selected dither and failure outcome, and it adds no error term. Each attempt is compared with its abort mass, so its error is not divided by an acceptance probability and has no second retry factor.

Conditional forgery bound for Pretzel-(3,3). In addition to the hypotheses of the comparison of real and simulated signing, assume the explicit-label extraction and conditions on the filter of Lemma C.2 of the Pretzel-(4,4) proof, specialized to these dimensions. Keys have CBD_1^3 secret and CBD_1^3 error, with the key-norm restriction and the dither selection handled by Paragraph H.2. The auxiliary instance has a uniform 3×4 relation matrix, or its adjoint in the auxiliary convention, with CBD_2^3 secret and CBD_6^4 error. The extraction norm is bounded by $\beta^2 = 4(1593^2 + 3068^2) = 47{,}801{,}092 < 7681^2$. The integer secant cap gives $\kappa_0 = 2^{964091/65536}$.

Let $\Delta_\mu = \min\{1, 400(Q+2)^3/2^{512}\}$ be the bound on message-prehash collisions. Use the notation λ_n, K_*, and λ_{MR} of Section F.4 for the success probabilities of finite selection, with trial cap $J \geq 1$. Use $P_{\mathrm{pair}} = 0$ for EUF. For SUF, keep $P_{\mathrm{pair}} = K_*(s_{\mathrm{pair}} + a_{\mathrm{key,pair}} + 3\delta_M)$, under the same condition on pairs of responses with equal challenges. For forgery probability ϵ, define $\epsilon_0 = [\epsilon - \Delta_F - \Delta_\mu - P_{\mathrm{pair}}]_+$ and use f_C from that section. Then the argument of Corollary F.5, specialized to these dimensions, gives

$$f_{2^{204}}\left(\frac{\lambda_{\mathrm{MR}}\epsilon_0}{(2q_B+1)^2}\right) \leq K_*\big[a_{\mathrm{key},0,+} + \kappa_0(s_{\mathrm{ext},+} + a_{\mathrm{aux},+} + e_R + e_U) + (2+3\kappa_0)\delta_M\big].$$

The advantages, the finite-circuit errors e_R, e_U, and the complete algorithm costs are as in Corollary F.5, with these dimensions and the oracle simulation error δ_M above. The proof follows the Pretzel-(4,4) proof step by step. Apply the bound for replacing signing by simulation, which includes aborts. Charge message collisions and any pair of responses with the same challenge. Then guess the attacked key with bounded trials and apply measure-and-reprogram with the 2^{204}-element challenge space. Use the retained 52-bit suffix construction of Appendix C.2: the first raw label is fresh and uniform on 256 bits, and the second extraction execution retains its suffix. The proof of Lemma C.2 applies unchanged to these dimensions and secant cap. Its coefficient multiplies both the extraction success and the auxiliary advantage; key replacement is charged separately.

Writing $\mathcal{B}$ for the complete right-hand side, $a = 2^{-204}$, and $L = \Delta_F + \Delta_\mu + P_{\mathrm{pair}}$ gives the explicit bound

$$\epsilon \leq \min\left\{1, L + \frac{(2q_B+1)^2}{\lambda_{\mathrm{MR}}}\left(a + ((1-a)\mathcal{B})^{1/3}\right)\right\}.$$

This follows from $(x-a)^3 \leq x(x-a)^2$ for $x \geq a$; for $x < a$ the same bound is immediate.

With the 2^{204}-element challenge family, the generic message search of Section 6, which keeps one valid signature and searches for another message with the same challenge, takes order 2^{204} classical trials or 2^{102} ideal quantum oracle evaluations. Table 5 gives lattice-attack estimates for the Pretzel-(3,3) key, auxiliary, and SIS instances, the last at the separate-cap radius $\sqrt{47{,}801{,}092} \approx 6913.8$. The key instance has integer dimension and sample count $(768, 768)$, modulus 7681, and unrestricted CBD_1 secret and error. The auxiliary instance has dimension and sample count $(768, 1024)$ with CBD_2 secret and CBD_6 error. Normal-form extraction SIS has $(n, m) = (768, 1792)$. These modeled work factors do not instantiate the advantage functions in the reduction; the theorem states those functions symbolically.

I Pretzel-(5,5) Parameters and Conditional Security Bounds

I.1 Pretzel-(5,5) parameters

Pretzel-(5,5) is the same Pretzel construction with module dimensions $(\ell, m) = (5, 5)$. We compare its attack estimates with ML-DSA-87. Table 9 lists the numerical changes from Pretzel-(4,4). Private randomness is generated with a pseudorandom function. Pretzel-(5,5) uses a wider mask (proposal) distribution, 64 quantizer labels, and a compressed signature encoding of 1908 bytes, specified below.

The base of the target distribution is the dyadic rational $\theta_{55} = T_{55}/2^{256}$, with hexadecimal numerator

$$T_{55} = \texttt{0xfffff0ed5689f6203b3b1e5aa8bcd882}\\ \texttt{b98cacf9f483e3c156cfb663778c15fb}.$$

The two displayed strings are concatenated.

Parameter	Pretzel
Module dimensions (ℓ, m)	$(5, 5)$
Estimated comparison target	ML-DSA-87
Signature bytes (compressed)	1908
Public-key bytes	2158
Quantizer levels	64
Key-norm statistic interval	[146,057, 307,135]
Weighted squared norm $E(s)$	[131,557, 292,635]
Envelope M	65,586/65,536
Private scalar precision (bits)	320
Key-candidate and attempt entropy (bits)	512
Private PRF key and public salt (bits)	256
Fine polynomial degree	57
Fine derivative rows	46
Signing attempt cap	512
Coefficient caps $(B_{\infty,h}, B_{\infty,r})$	$(288, 450)$
Euclidean caps $(B_{2,h}, B_{2,r})$	$(2150, 3170)$
$\log_2 \kappa$	$40{,}741{,}973/2^{20}$
Extracted SIS bound β^2	58,848,400

Table 9: Pretzel-(5,5) parameters. As in Pretzel-(4,4), $n = 256$, $q = 7681$, the weights of E are $(9216, 196, 121)$, and the challenge set has 2^{255} elements. The SIS bound conservatively rounds up the extraction bound. The fine polynomial is the longer approximation used to check deferred transport steps (Section I.2). Its coefficients for each key are computed from a table whose rows hold the Taylor coefficients of successive terms of the series it approximates. The number of rows appears as "fine derivative rows" in the table. Section I.2 gives the work and transfer certificate for the encoding of 1908 bytes.

The envelope M must exceed $\mathcal{M}(b)$ for every allowed value of $b = E(s)$. The monotonicity argument in Appendix A applies to every fixed $0 < \theta < 1$, so it suffices to bound $\mathcal{M}$ at the largest allowed value. For θ_{55}, outward repeated squaring at scale 2^{768} and the alternating partial sums through terms 20 and 21 enclose the theta series from above and below, respectively. Its decreasing terms bound the omitted tail. Dividing outward by an enclosure of θ_{55}^{292635} and adding one gives $65{,}585/65{,}536 < \mathcal{M}(292{,}635) < 65{,}586/65{,}536$, so the envelope covers the full interval of weighted squared norms $E(s)$. A larger envelope lowers the ideal probability that an attempt survives all 128 transport positions, including those used for dummy tests. Actual acceptance also depends on the public checks applied after transport.

The key-norm statistic is $23{,}716 + 196\|s_0\|_2^2 + 121\|e\|_2^2$, while $E(s) = 9216 + 196\|s_0\|_2^2 + 121\|e\|_2^2$, so the two intervals in Table 9 differ by 14,500. Pretzel-(5,5) uses a wide key-norm interval to reduce the error of replacing the restricted key distribution in the security proof.

Response bounds. The signature contains the salt, the first two response blocks (the *head*), and information identifying the commitment labels. Every head coefficient must lie in $[-288, 288]$, and the verifier applies the public norm and secant bounds in Table 9. For the value x that the verifier recomputes, label $u \in \{0, \ldots, 63\}$ gives the residual $\mathrm{ctr}_q(c_u - x)$, where

$$c_u = 120u + 60 + \mathbf{1}_{u \neq 0 \wedge u \text{ even}}.$$

These are the quantizer centers. The residual must have absolute value at most 450. The challenge hash takes the absolute labels as input. The extraction norm satisfies $4(2150^2 + 3170^2) = 58{,}685{,}600 < 58{,}848{,}400 < q^2$, so the attack estimates can use the larger SIS bound in the table.

Compressed encoding. Pretzel-(5,5) signatures have a fixed length of 1908 bytes, including a 32-byte salt. The encoder always emits exactly this length, and the verifier rejects any other length. As in Pretzel-(4,4), the head is sent as low parts with a short list of exceptions, and the labels as local indices relative to the value that the verifier recomputes. The first 256 head coefficients have low parts of six bits in two's complement, in $[-32, 31]$. The other 1280 have low parts of eight bits in $[-128, 127]$. A count byte and at most 40 entries of twelve bits each list the coefficients outside these ranges. Each entry contains a position of 11 bits and a sign bit. It adds ± 64 to a coefficient among the first 256 and ± 256 to any other. Each label is sent as a local index $r \in \{0, \ldots, 7\}$: two low bits per label, then a count byte and a list of at most 16 positions, each encoded in eleven bits. At each listed position, the label index is increased by 4. Positions in both lists are strictly increasing and unused entries are zero. The verifier rejects head positions outside $[0, 1536)$, label positions outside $[0, 1280)$, and counts above 40 and 16, respectively. The verifier maps its recomputed value x and the index r to a label through the public recovery row of Lemma D.1. For each x, list all labels u with $|\operatorname{ctr}_q(c_u - x)| \le 450$, ordered by $(|\operatorname{ctr}_q(c_u - x)|, \operatorname{ctr}_q(c_u - x))$. The seven or eight entries are distinct; a negative residual precedes its positive counterpart at a tie. Reject an index without an entry. The challenge hash takes the recovered labels as input. The fields occupy $32+192+1280+1+60+320+1+22 = 1908$ bytes and have no padding bits, and each signature value has exactly one accepted encoding. Decoding rejects a coefficient among the first 256 outside $[-95, 95]$ and any other head coefficient outside $[-288, 288]$, and applies the public norm, residual and secant checks above. A response that does not fit these fields is rejected before encoding, and the attempt restarts.

Canonical decoding. This paragraph completes the proof of Lemma 4.6 for the 1908-byte encoding. For Pretzel-(5,5), a decoded coefficient determines its two's-complement low part uniquely. Its difference from that low part is either zero or exactly one of ± 64 (leading) and ± 256 (nonleading), so it determines the presence and sign of its exception entry. Strictly increasing positions and zero unused entries make the list of entries unique, with head positions outside $[0, 1536)$, label positions outside $[0, 1280)$, and counts above 40 and 16, respectively, rejected. The distinct entries in each public recovery row determine each local index uniquely; its high bit determines membership in the label-exception list. Thus the same inverse identities hold under the stated coefficient bounds and limits on the numbers of exceptions.

Public and secret keys occupy 2158 and 782 bytes, respectively. The Module-LWE and Module-SIS instances, extraction bound β^2 and loss factor are those of Theorem I.1. The encoding's additional checks on coefficient values and whether the response fits in the encoding only restrict the response set defined by the public bounds above, so Proposition D.4 preserves the extraction and secant bounds. No new hardness assumption is needed, and the Pretzel-(5,5) security theorem is the cubic Theorem I.1.

Bounded dither selection and its scope. After fixing the public core, Pretzel-(5,5) draws a 32-byte candidate-start seed; failure of this entropy request returns an error. Candidate i, for $0 \le i < 2048$, adds i to the candidate-start seed as a little-endian integer modulo 2^{256}. Its bounded expansion reads at most 1664 little-endian 16-bit words, keeps the low 13 bits when they are below 7681, and succeeds upon obtaining 1280 values. Return the first candidate seed whose expansion succeeds, serialized with the core. Only expansion exhaustion advances to the next candidate; any other error is terminal. If no candidate expands, return failure without resampling the core. No public sample responses are drawn or tested. Public-key preparation expands the serialized seed.

The following bounds concern this public continuation for every canonically encoded core, with successful entropy in the ideal-XOF experiment. They do not bound preceding core generation or entropy failure.

The reference row for Pretzel-(5,5) is the analogue of the reference sample of Section H.1, with the level-five base, weights and centers. It uses 2816 words: 256 for the leading, 1280 for the other head and

1280 for the error coordinates. With $U = Q_{q,64}(A_h y_h + y_e + \delta)$ and $r = \mathrm{center}_q(C(U) - A_h y_h - \delta)$, success requires every coordinate to exist and (y_h, r) to pass all level-five coefficient, Euclidean, secant and canonical 1908-byte encoding checks. The finite row certificate gives the following upper bounds on the union of failed samples, coefficient caps, capped head norm, capped residual norm, capped secant score, and 1908-byte encoding limits, respectively:

$$\begin{aligned} &0.000221431255340577, \quad 0.000067651271820069, \\ &3.393103608912 \cdot 10^{-9}, \quad 1.546708057555 \cdot 10^{-14}, \\ &0.010293822690649243, \quad 0.000085589061162208. \end{aligned}$$

Their sum is below $0.0106684976720912 < 11/1000$, uniformly in the fixed core, dither and prescribed parity. These numerical enclosures certify the finite public row predicate, including the 1908-byte encoding limits.

The moment argument conditions on the entire head, which fixes every residual context. Omitted coordinates remain independent, and each capped residual moment is bounded by its maximum over all contexts. Multiplying before removing the conditioning and charging missing mass as failure gives the stated union bound. In particular the finite row success probability exceeds $989/1000$ without renormalizing the rows. Put

$$d = \Pr[\mathrm{Bin}(1664, 511/8192) \geq 385].$$

Candidate zero fails bounded expansion with probability d. Failure of all candidates implies failure of candidate zero, so the continuation fails with probability at most $d < 4.93260021844487 \cdot 10^{-111}$. This uses no independence between candidate expansions. Pointwise domination of the finite row masses by the full Gaussian masses gives $p_{\mathrm{ideal}} \geq p_{\mathrm{finite}} > 989/1000$ for every fixed core, dither and prescribed parity. This is a bound for the public row predicate, not for private transport or a complete signing call.

Theorem I.1 proves its bound directly for this setup. Both sides use the same total public continuation, including its selected seed and failure outcome.

Section I.2 gives complete work bounds for all seven algorithms of Theorem I.1 for Pretzel-(5,5). It also transfers the theorem's hypotheses for finite sampling distributions, independence and access conditions for private randomness, advice, and finite oracle simulation. The original functional and analytic assumptions remain in force, with no new hardness assumption. The work bounds establish a finite reduction. They do not supply a new concrete estimate of security at category 5. The parameter-set identifier appears in both key envelopes, in the key-seed hash input, and in the complete public message and challenge hash inputs. The private salt, proposal and transport hash inputs do not contain a separate parameter-set field. Their binding to this fixed parameter set follows from the comparison with streams indexed by key-generation call in Section I.2, without an additional error term. The key tag is not authenticated, so each key must be used with one format only. Giving a key body a second parameter-set identifier reuses its private PRF key, which identifies its private streams, and would repeat the salt, proposal and transport streams, so this rule of one format per key also keeps the private streams of different parameter sets apart.

The signatures of 1908 bytes are smaller than SHUTTLE-255's signatures of 2013 bytes. This comparison concerns sizes only. We do not compare speed across the two studies' platforms.

The public key consists of a 14-byte parameter-set identifier, a 32-byte matrix seed, 1280 public coefficients at 13 bits each, and a 32-byte dither seed. Thus $|\mathsf{pk}| = 2158$. Verification checks canonical encoding and the public bounds for Pretzel-(5,5) in Table 9.

I.2 Conditional finite reduction and work for Pretzel-(5,5)

The larger parameter set changes the sampler's dimensions and error bounds. This subsection bounds these errors in the same simulation experiment as for Pretzel-(4,4), including key-generation failures and reuse of

the cached result when an attempt name repeats. Let K count key-generation calls, including failed ones, and let $A_{\text{att}} \le 512S$ count signing attempts in S signing calls. For parameter set $j = 55$, let D_j be the statistical distance between the output distribution of one finite signing attempt, including aborts, and that of its ideal counterpart, without conditioning on success. Let η_j be the probability that one key candidate violates the key-norm restriction. Under Assumption F.3, the bounds used below are

$$D_{55} < 2^{-304}, \qquad \eta_{55} < 2^{-289}.$$

The bound on D_{55} combines certificates for the polynomial approximations, the conversion of each step's random word into an integer logarithm described below, the mask sampler, the lists of pending values, and the dummy tests. It includes abort outcomes. It requires the specified invariants on valid states and that the implementation's arithmetic equals its mathematical specification.

Scope of the sampling comparison. The observable result of an attempt is its returned signature or abort, together with the state kept for later calls. A rejected proposal, temporary lists of pending values, and instruction or memory-access traces are not observable in this experiment. Reset must discard a failed attempt's private state before the next attempt.

Deferred step decisions. Each Pretzel-(5,5) transport step uses an independent 384-bit random word. As described in Section 4.3, the implementation first locates the logarithm of the scaled coin between consecutive integers, refining close boundaries, and then tests whether the resulting integer lies in a union of integer intervals. A separate short (coarse) polynomial evaluates the correction comparison. An active step is *marked* when that comparison lies in its uncertainty band; it is applied provisionally and checked at the end of the attempt with a longer (fine) polynomial. As in Appendix D, the plus and minus choices below are swapped when the pre-update correlation is negative. This deferral has a direct path-by-path justification.

At a transport step, let P be the add test on the step's word (the union-of-intervals test) and F the subtract test computed with the fine polynomial on the same word. Deciding the step immediately, the finite transition chooses plus if P, minus if $\neg P \wedge F$, and aborts otherwise, followed by the common sign convention. A marked step instead provisionally chooses plus if P and minus otherwise. It saves the inner product $\langle z, X^j s\rangle_W$ before the update (the step's *correlation*) and the same random word, and the final check rejects exactly when $\neg P \wedge \neg F$. For unmarked steps, the uniform error bound of the coarse polynomial must ensure that its decision agrees with F.

Hence, on every run in which no deferred check fails, each provisional update equals the immediate update, and induction over the steps proves that all correlations and the returned signature or abort, together with the state kept for later calls agree. Conversely, a successful immediate run has $P \vee F$ at every step, so the deferred version with unlimited storage for deferred computations makes the same updates and passes every final check. On failed runs, both versions return the same abort and reset state. Limits on this storage can cause an additional abort, charged by the bounds on exceeding those limits. When these hypotheses hold, provisional updates need no separate charge.

The bound on storing too many marked steps allows the correlations to depend adaptively on earlier steps. Before step i, the attempt's mathematical state depends only on the complete words of earlier steps. For every such state the current word is uniform, and the step is marked with probability at most $p_c = 2{,}145{,}846{,}757/2^{64}$ for the larger Pretzel-(5,5) envelope. Conditioning step by step bounds the probability that any fixed eleven steps are all marked by p_c^{11}. A union bound gives $\binom{128}{11} p_c^{11}$ for needing to store more than ten marked steps. This proof does not require the marking events to be independent. After an abort, the process can be continued from a fixed valid state. Failures in computing the integer logarithms (more ambiguous conversions than can be stored for recomputation, or a recomputation that remains ambiguous) are charged separately, and the estimate for storing marked steps does not condition on all integer logarithms being computed successfully.

Correlations outside their valid range, and integer logarithms outside the range on which their computation is certified, are excluded by explicit hypotheses on the implementation, rather than assigned an unproved small probability.

Sampling the Pretzel-(5,5) mask. For each of the 2560 nonleading coordinates, the first lookup directly selects a magnitude from 0 through 254, or a symbol requesting the tail. A tail lookup applies the same rule to the next 255 magnitudes. Six public levels cover the supports, 1149 for weight 196 and 1462 for weight 121. Each lookup has 256 columns and a 320-bit comparison threshold. An independent fair sign bit converts magnitudes into signed values. An exact calculation reconstructs the probabilities of the sampler's finite paths and bounds their distance from the ideal Gaussian, including directed rounding errors in the masses and omitted tails.

The implementation first compares a 16-bit prefix. It packs coordinates that this prefix leaves unresolved, or that request the tail, into a list of at most 64 coordinates to refine in their original order. Refinement keeps that prefix, extends it to 48 bits, and supplies the next column and a 32-bit prefix for the second level. A second list holds at most eight coordinates, each with a 512-bit random word for final refinement through all six levels. Exceeding either list size or exhausting a random word aborts the whole mask sample. Every prescribed position is processed.

For the analysis, assign an independent random word to every coordinate before any coordinate is selected for refinement. Selection depends only on previously revealed prefixes, so assigning the unused physical random words in stable order gives the same distribution until a list exceeds its size limit. The certificate counts the relevant prefix cells exactly. Union bounds over subsets of 65 and 9 coordinates bound the probabilities of exceeding the first and final list sizes by 2^{-428} and 2^{-313}, respectively, without conditioning on either size limit being met.

Each final comparison consumes only the shortest prefix that decides it; if the full word equals the threshold, the comparison is decided at the end of the word. Conditional on that prefix, the unused suffix remains uniform, so it can supply a subsequent column and comparison. This argument concerns only consumed prefixes; it does not assume that the implementation's overlapping lookahead windows are independent. The number of fresh bits a comparison consumes is conditionally dominated by a geometric random variable with success probability $1/2$. There are at most six comparisons and four new column bytes, so a union bound over the eight random words bounds the probability that any random word is exhausted by

$$8\,2^{-480}\sum_{j=0}^{5}\binom{480}{j} < 2^{-439}.$$

Adding these abort probabilities and the finite Gaussian error to the terms for the leading-coefficient sampler and transport gives, under the stated hypotheses,

$$D_{55} < 0.024330647\,2^{-304} < 2^{-304}.$$

The mask and transport tapes are a fixed partition of two 8544-byte XOF outputs with distinct encoded inputs. The first supplies 8544 mask bytes, and the second supplies the remaining 2400 mask bytes followed by 6144 transport bytes. Independent uniform XOF outputs therefore give independent uniform tapes. The oracle accounting charges complete bounded-prefix queries and their inverses, separately from the implementation's permutation count. The implementation must satisfy the theorem's naming and query-budget hypotheses for these inputs and prefix lengths. The bound also requires that refinement preserves prefixes, that random words are assigned in stable order, that aborts and resets behave as specified, and the other hypotheses for replacing the implementation's randomness. Because it does not condition on success, it composes with the attempt count below.

The attempts contribute $A_{\mathrm{att}} D_j$, and replacing the accepted keys contributes $K\eta_j(1-\eta_j^{512})$. For example, for $K \le 2^{32}$ and $A_{\mathrm{att}} \le 2^{41}$, the per-attempt and per-candidate bounds above give a sum below $2^{-263}+2^{-257} = (65/128)2^{-256}$. These are only the statistical sampling terms. The full bound for replacing signing by simulation also includes the replacement of private hash families, selective programming of the challenge hash, the message-prehash comparison, and the error of the finite circuit that simulates the matrix random oracle.

These terms follow from the key-replacement and sequential-composition arguments of Appendix E. The first accepted key has the distribution of one candidate conditioned on the key-norm restriction, and key generation fails with probability η_j^{512}; replacing the accepted key on the success branch costs $\eta_j(1-\eta_j^{512})$. The attempt comparison includes aborts and composes over first occurrences of attempt names, without dividing by an acceptance probability.

The public response sampler. A finite public response sampler gives $e_{\mathrm{pub},j} < 2^{-326}$ for Pretzel-(5,5). It computes normalizers as intervals with directed rounding at precision 640 and rounds cumulative probabilities to a 352-bit grid. All Gaussian coordinates are truncated at absolute value 4096, and the omitted normalized mass, below 2^{-1933} per coordinate, is charged explicitly. The public map to signature bytes and its checks are applied after this sampling step. The tail charge also covers ideal values outside the input range assumed by the implementation's modular reduction.

The Gaussian tail bound has a direct conservative proof. For any of the weights $w \in \{9216, 196, 121\}$, put $a = -w\ln\theta$. For the selected Pretzel-(5,5) parameter, $121(1-\theta) > 1/9216$. Since $-\ln\theta \ge 1-\theta$, we have $a > 1/9216$. The normalizer of an unrestricted or even-parity coordinate is at least one. For an odd-parity coordinate it is at least $2\theta^w > 1$, since $\theta \ge 1-2^{-19}$ and $\theta^w \ge 1 - w2^{-19} > 1/2$. Bounding either parity tail by the unrestricted unnormalized tail gives

$$\Pr[|X| > 4096] \le \frac{2e^{-a4097^2}}{1-e^{-8195a}} < 4e^{-1800} < 2^{-2398} < 2^{-1933}.$$

The geometric denominator follows because successive squares in the tail differ by at least 8195. Here $8195/9216 > 3/4$, $4097^2/9216 > 1800$, and $e^{3/4} > 2$ follows from its first five Taylor terms. Thus the stated tail allowance holds before any public filtering or conditioning on signing success. At most 2816 coordinates occur in Pretzel-(5,5), so replacing their joint distribution by its coordinatewise truncation costs less than 2^{-2386} by product coupling. This tail term is separate from the CDF-rounding and directed-normalization errors in $e_{\mathrm{pub},j}$.

The following construction with directed rounding suffices for the CDF part of the public sampler. Each retained weight is θ^E with $E \le 9216 \cdot 4096^2 < 2^{38}$. Evaluate powers by binary repeated squaring with outward rounding to 640 fractional bits, clipping all intervals to $[0,1]$. The width of an interval product is at most the sum of its input widths plus $2 \cdot 2^{-640}$. Induction on the power computation therefore bounds a weight interval's width by $2(E-1)2^{-640} < 2^{-601}$ for $E \ge 1$; exponent zero is exact. For $s \le 8193$ retained values, exact addition gives total interval width $W < 2^{-587}$. The true normalizer is at least one, as above, so its lower endpoint is greater than $1 - W > 1/2$. Divide each cumulative numerator interval by the total interval with outward 640-bit rounding. Its width is at most $4W + 2 \cdot 2^{-640} < 2^{-580}$. Clipping to $[0,1]$ preserves enclosure.

Let F_i be the exact truncated CDF. Round its certified lower endpoint down to the 352-bit grid, take running maxima to ensure monotonicity, and set the last threshold to one. The resulting thresholds G_i satisfy $0 \le F_i - G_i < 2^{-352} + 2^{-580}$ at every internal boundary. Indeed, a running maximum stays below F_i because the exact CDF is nondecreasing. Sampling with a uniform 352-bit integer gives masses $G_i - G_{i-1}$. By the

triangle inequality on adjacent CDF errors, its statistical distance from the truncated distribution is at most

$$\frac{1}{2}\sum_i |(G_i - G_{i-1}) - (F_i - F_{i-1})| \le \sum_{i=1}^{s-1} |G_i - F_i|.$$

There are at most $2816 \cdot 8192 < 2^{25}$ internal boundaries over all coordinates. Their total CDF error, including the tail bound, is less than $2^{-327} + 2^{-555} + 2^{-2386}$.

The public response experiment also draws its independent acceptance bit with probability $\alpha = (Z_o/(MZ_e))^{128}$, as in Appendix E. Its normalizers use the same 640-bit directed intervals. Each has interval width below 2^{-587}, lower endpoint above $1/2$ and upper endpoint below 2^{14}. Thus the ratio interval, clipped to $[0, 1]$, has width below 2^{-567} after including the omitted tails. Seven squarings amplify that width by at most 2^{14}; their outward rounding leaves width below 2^{-550}. Rounding the lower threshold to 352 bits and comparing with an independent uniform 352-bit integer costs statistical error below $2^{-352} + 2^{-550}$. The joint sampling of the coordinates and independent acceptance bit has

$$e_{\mathrm{pub},j} < 2^{-327} + 2^{-555} + 2^{-2386} + 2^{-352} + 2^{-550} < 2^{-326}.$$

The public conversion to signature bytes and every failure outcome are applied after this sampling; no conditioning on acceptance is used. The work for the threshold and its draw is included in Section I.2.

Fixed tapes and the matrix oracle simulator. Pretzel-(5,5) uses fixed-length mask and transport tapes. Their overflow and exhaustion terms are already included in D_{55}, so no monitor for variable-length draws is needed and $\tau_{55} = 0$. As for Pretzel-(4,4), the comparison uses the functional matrix interface: the simulator returns the matrix directly and answers bounded raw-prefix queries with the joint distribution of the matrix and the raw stream. The analysis gives no worst-case bound on the running time of the implementation's matrix rejection sampling.

The matrix has 6400 coordinates. The simulator places them in the first 3328 and 3072 accepted values of its two 13-bit streams, respectively, with a bijective coordinate assignment. At matrix-query budget Q_{mat}, the oracle simulator uses $12{,}800Q_{\mathrm{mat}}$ base-field coefficients. Truncating each coefficient's rejection sampler after t_c trials contributes at most that count times $(511/8192)^{t_c}$. Over the same degree-20 extension field as for Pretzel-(4,4), the oracle simulator uses $6400/20 = 320$ polynomials of degree $2Q_{\mathrm{mat}} - 1$, with no unused coordinates. Each polynomial has $2Q_{\mathrm{mat}}$ extension-field coefficients, each represented by twenty base-field coefficients, which gives the stated count.

The public salt remains 256 bits. Accordingly, the term for selectively programming the challenge hash is still

$$e_C \le \min\left\{1, \frac{6NQ}{2^{128}} + \frac{\binom{N}{2}}{2^{256}}\right\},$$

where N counts first occurrences of complete attempt names, including attempts that abort without programming a point, and Q is the total budget of charged complete-input queries. Pretzel-(5,5)'s 64-byte private entropy does not change this public-salt term.

Feature precision for the larger dimension. The finite-feature construction of Appendix B.4 specializes with $d = 256(\ell + m + 1)$ response coordinates, namely $d = 2816$. The coefficient domains and feature exponents keep the per-coordinate degree-tail bound $\Pr[K > D] \le 2^{1-D}$. For an integer $B \ge 0$ and requested error 2^{-B}, take

$$D = 2B + 20, \qquad g = \lceil \log_2(T + 1) \rceil, \qquad R = 2\big(d(D + 1) + g\big), \qquad p = B + \lceil \log_2(32R) \rceil,$$

where T is the secant threshold of Pretzel-(5,5), in units of 2^{-20}. Thus $g = 26$, so the larger threshold requires one more binary-gap attenuation flag than Pretzel-(4,4). Since $d < 4096$, truncation changes the feature direction by less than $\sqrt{d\,2^{2-D}} < 2^{-B-3}$. The angle and synthesis errors total at most $2R2^{-p} \le 2^{-B-4}$. The same disjoint completion spaces, exact sign phases, and controlled-state argument therefore give $\varepsilon_{\mathrm{iso}} < 2^{-B}$ on arbitrary side information. The lookup work becomes $O(d \cdot 741Dp)$, with uniform table generation and gate synthesis charged separately.

The other two circuit errors can share an explicit budget with these features. For a combined per-experiment allowance 2^{-B}, use the feature schedule with requested precision $B + 2$. Sample each auxiliary mask coefficient by reducing $B + 24$ independent fair bits modulo q. The total variation bound from Section 5.2 gives

$$\delta_u \le \frac{256mq}{4 \cdot 2^{B+24}} \le \frac{320 \cdot 7681}{2^{24}} 2^{-B} < 2^{-B-2},$$

where $m \le 5$. For the phase, compute the exact residue $j = [v, \xi]_q$ reversibly, apply the phase indexed by j, and uncompute the residue workspace. Rounding j/q to the nearest multiple of $2^{-(B+4)}$, modulo one, incurs phase-angle error at most $\pi 2^{-(B+4)} < 2^{-B-2}$. Allow at most 2^{-B-2} additional operator error for the complete synthesized lookup-controlled phase block, including all of its elementary gates. The inequality $|e^{ix} - e^{iy}| \le |x - y|$ and the retained response labels then give $\varepsilon_\phi < 2^{-B-1}$ uniformly, including reference registers. Consequently

$$\varepsilon_{\mathrm{iso}} + \varepsilon_\phi + \delta_u < 2^{-B}.$$

The isometry and phase bounds imply the same per-experiment bounds on output probabilities, by the trace-norm comparison in Appendix B.4. The classical mask error bounds the change of the resulting channel even with a reference register. Thus Theorem I.1 charges their sum as $2\kappa_j(\varepsilon_{\mathrm{iso}} + \varepsilon_\phi + \delta_u)$ inside the target-selection factor K_*. Here 2^{-B} is a budget for circuit approximation errors, not a forgery-probability bound. The uniform construction and cost of the phase block are charged as in the general circuit hypotheses.

Finally, the extraction theorem specializes with the table's β and κ. The key Module-LWE instance has dimensions $(5, 5)$ with CBD_1 secret and noise. The auxiliary instance has dimensions $(5, 6)$ with CBD_2 secret and CBD_6 noise. The challenge map remains a bijection from 255 uniform input bits, so every Pretzel-(5,5) challenge has probability 2^{-255}. The extraction function $f_{2^{255}}$, the advice restrictions, the query charges, and the isometry hypotheses are unchanged.

The work bounds use the functional and analytic hypotheses of Theorem I.1 and Assumption F.1, and cover the specified matrix interface. Evaluate the hardness advantages at the constructed algorithms' actual costs.

Cost model and terms. Count NOT, CNOT, Toffoli, H, T, T-dagger, preparation and measurement as one gate each. All table access, arithmetic, constant generation and reversal are charged. There is no QRAM or free arbitrary-precision arithmetic. Use the theorem's query budgets, padded input and output widths I, L, selection cap J, and precision parameter B. Let W_A be the adversary's nonoracle work, including preprocessing, W_{prep} the cost of preparing its independent advice, and S_A its initial workspace, including its one original advice state.

The following terms denote explicit upper bounds. F bounds one public computation, including key preparation, commitment, filtering, encoding or decoding, and the final lattice checks. C_{setup} and C_{sign} bound one complete setup and signing call, including failures and retries.

For Pretzel-(5,5), C_{setup} covers core preparation, the entropy request, up to 2048 bounded dither expansions, selection, serialization, retained outcomes and cleanup under Section I.1. It includes no generation, filtering or caching of public test rows. The independent key search used for loop control in the proof is also charged. The public response sampler used to compare signing distributions is a separate proof computation and keeps its full work and query charges. C_{shared} covers global constant generation, saved outputs and

state changes and query adapters. C_{final} covers the final acceptance, freshness, identity, lattice and readout checks. $\mathrm{Sel}(n, J)$ bounds selection among n values with at most J trials, retaining trial bits and failure outcomes. Finally, $E_f(B)$, $E_a(B)$ and $E_u(B)$ bound feature preparation, the auxiliary phase and auxiliary mask sampling, with their combined error below 2^{-B}.

Write $W_p = W_{\mathrm{bin}} + W_{\mathrm{mat}}$ for the total work of the finite binary and matrix oracle simulators. This includes initialization, random bits, all calls and their inverses. Their state persists across executions and final tests. The query budgets already cover three executions, so W_p is charged once. Define

$$\begin{aligned} C_{\mathrm{run}} &= KC_{\mathrm{setup}} + SC_{\mathrm{sign}} + 8VF, \\ C_3 &= C_{\mathrm{shared}} + 3C_{\mathrm{run}} + \mathrm{Sel}(K, J) + \mathrm{Sel}(2q_B + 1, J) + C_{\mathrm{final}}, \\ C_1 &= C_{\mathrm{shared}} + C_{\mathrm{run}} + \mathrm{Sel}(K, J) + C_{\mathrm{final}}, \\ C_{\mathrm{aux}} &= C_3 + E_f(B) + E_a(B) + E_u(B) + 8F + 2^{24}. \end{aligned}$$

Selection does not rerun the adversary K times. Neither K nor κ_{55} is an execution multiplier.

Theorem I.1 (Conditional finite specialization for Pretzel-(5,5))**.** Fix Pretzel-(5,5) and put $j = 55$. Use the security experiment of Section F.1 and the interfaces of Appendix E for replacing signing by simulation, with this parameter set's sampler and encoding, and the total setup without public tests of Section I.1. The setup clause of Assumption F.1 uses that candidate order, bounded expansion and failure interface. Each successful entropy request returns fresh uniform bytes, and entropy failures behave the same way in both experiments, independently of the secret. Assume that the implementation meets its integer, buffer, transport, cap, and encoding contracts, and assume the explicit-label isometry and extraction hypotheses for these dimensions. Replacing the key restricted by the key-norm interval with an unrestricted key uses the witness-free continuation of Appendix E. The finite-sampling comparison uses Assumption F.3 for Pretzel-(5,5), together with the analytic identities and valid-state invariants of its certificates. These include the uniform series and derivative bounds for the fine polynomial and the bound comparing the coarse and fine polynomials. Each routed correlation must be exact and lie in its certified range; the implementation obtains it from the routed initial inner product and earlier updates, as described in Section 7.2. Each integer logarithm must also lie in its certified range. The conditions on saved data and reset of Section I.2 apply: the state before a step depends only on earlier random words, a marked step saves its pre-update correlation and random word, and reset discards a failed attempt's private state.

Let $1 \le K \le 2^{64}$ bound the number of key-generation calls. Use S, V, h, r, $A_{\mathrm{att}} = 512S$, and $N \le A_{\mathrm{att}}$ as in Section F.1, keeping the outputs and state changes of rejected and repeated attempts. The conservative query budgets are

$$\begin{aligned} R &= r + 7448K + 32A_{\mathrm{att}} + 16V + 8S + 32, & D_M &= 12K + 8(V + S + 1), \\ Q_{\mathrm{bin}} &= 180R + 6h + 8 + 6A_{\mathrm{att}}, & Q_{\mathrm{mat}} &= 96R + 6D_M, \\ Q &= Q_{\mathrm{bin}} + Q_{\mathrm{mat}}, & q_B &= 2h + 2r + 2V + 2S + 2. \end{aligned}$$

These budgets count both evaluation and uncomputation, including two calls for each direct query to the programmed challenge oracle. Let $b = 512$. For u independent b-bit draws, write $d_b(u) = 1 - (2^b)_u/(2^b)^u$ for the probability of a collision, where $(n)_u$ is the falling factorial and $(n)_0 = 1$. Take $d_b(u) = 1$ when $u > 2^b$. Define

$$\begin{aligned} \delta_G &= d_{256}(K), & e_P &= \min\{1, 4KQ^2/2^{256}\}, \\ e_{KD} &= d_b(512K) + (1 - d_b(512K)) \min\{1, 2048KQ^2/2^b\}, \\ e_C &= \min\{1, 6NQ/2^{128} + \tbinom{N}{2}/2^{256}\}. \end{aligned}$$

With the same bounded dither selection, including failure, run in both experiments, put

$$\begin{aligned} E_F &= e_P + e_{KD} + e_C \\ &\qquad + A_{\mathrm{att}}(D_j + e_{\mathrm{pub},j} + \tau_j) + K\eta_j(1-\eta_j^{512}), \\ \Delta_F &= \min\{1, \delta_G + (1-\delta_G)\min\{1, E_F\}\}, \end{aligned}$$

where $\tau_{55} = 0$ and the preceding certificates specify the other quantities. The comparison keeps the actual selected seed and its expansion, including key-generation failure. For a positive integer cap t_c on coefficient-sampling trials, let

$$\begin{aligned} \Delta_\mu &= \min\{1, 400(Q+2)^3/2^{512}\}, \\ \delta_M &= \min\{1, c_j Q_{\mathrm{mat}}(511/8192)^{t_c}\}, \qquad c_{55} = 12{,}800. \end{aligned}$$

Use the finite-selection notation λ_n, K_*, $\lambda_{MR} = \lambda_{2q_B+1}$, and the function f_C of Section F.4, with trial cap $J \geq 1$. Let $P_{\mathrm{pair}} = 0$ for EUF and $P_{\mathrm{pair}} = K_*(s_{\mathrm{pair}} + a_{\mathrm{key,pair}} + 3\delta_M)$ for SUF. If the forgery probability is ϵ, set $\epsilon_0 = [\epsilon - \Delta_F - \Delta_\mu - P_{\mathrm{pair}}]_+$. Then the finite lattice algorithms constructed by the reduction satisfy

$$\begin{aligned} f_{2^{255}}\left(\frac{\lambda_{MR}\epsilon_0}{(2q_B+1)^2}\right) \leq K_*\Big[&s_{\mathrm{ext}} + a_{\mathrm{key,rel}} + a_{\mathrm{key},0} + a_{\mathrm{key},1} + \kappa_j a_{\mathrm{aux}} \\ &+ (7+2\kappa_j)\delta_M + 2\kappa_j(\varepsilon_{\mathrm{iso}} + \varepsilon_\phi + \delta_u)\Big], \end{aligned}$$

Here the a terms are the distinguishing advantages of the constructed key and auxiliary Module-LWE algorithms, and the s terms are the success probabilities of the constructed SIS algorithms, from extraction or from a pair of responses with equal challenges. Their dimensions, distributions, norm bounds, and κ_j are those stated above. The three remaining errors are exactly the isometry, phase, and rounding errors of Section I.2. All algorithms obey the advice restrictions of Section F.1 and share the same state of the finite oracle simulators.

Proof. Use the proof of Theorem 5.6 with the following setup substitution. In every experiment run the total public selection of Section I.1, including its failure outcome. Its dither inputs lie outside the challenge domain, so challenge programming preserves its outputs and saved state. It reads neither the private PRF key nor private signing streams; isolating and replacing those families therefore leaves setup intact. The key-candidate replacement covers all 512 candidate labels. After the witness is removed, apply the same public selection to the external core without restarting or conditioning it. The common continuation argument of Appendix E then gives $K\eta_j(1-\eta_j^{512})$, including the all-failed branch. Within a setup call the matrix is fixed before this continuation runs, so setup cannot resample the external matrix. No empirical acceptance event is used in these steps.

For each fixed resulting core and dither, the signing comparison and canonical response filter apply with their existing abort mass and retained reference systems. The oracle simulation uses the conservative budgets counted below. These observations give the same proof with the Pretzel-(5,5) parameters: the query budgets above, the finite sampling bounds and auxiliary dimensions of Section I.2, and the response caps and κ_j of Section I.1. The setup substitution preserves the form of the error bound. The values of δ_G, e_P, e_{KD}, e_C and Δ_F are computed above. Four further inputs differ: e_{KD} uses 512-bit key-candidate labels; e_C is the coarser bound of Appendix E that charges every counted query; the per-attempt term D_j includes the arguments for deferred checks and their storage limits of Section I.2, with $\tau_j = 0$; and the canonical lift uses the 1908-byte decoding of Section I.1 with Proposition D.4, while Section I.2 transfers the hypotheses on independence and access to private randomness, advice, and finite oracle simulation. □

Complete work bounds. The resulting work bounds for all seven algorithms are

Algorithm	Complete work bound
s_{ext}	$W_{\mathrm{prep}} + 96W_A + 24S_A + W_p + 32C_3$
$a_{\mathrm{key,rel}}$	$W_{\mathrm{prep}} + 96W_A + 24S_A + W_p + 32C_3$
$a_{\mathrm{key},0}$	$W_{\mathrm{prep}} + 96W_A + 24S_A + W_p + 32C_3$
$a_{\mathrm{key},1}$	$W_{\mathrm{prep}} + 96W_A + 24S_A + W_p + 32C_3$
a_{aux}	$W_{\mathrm{prep}} + 96W_A + 24S_A + W_p + 32C_{\mathrm{aux}}$
s_{pair}	$W_{\mathrm{prep}} + 32W_A + 8S_A + W_p + 32C_1$
$a_{\mathrm{key,pair}}$	$W_{\mathrm{prep}} + 32W_A + 8S_A + W_p + 32C_1$

For each row, if W_{total} is its displayed work bound, storage is at most

$$S_A + 8(I + L) + 2^{24} + W_{\mathrm{total}}.$$

The work charges new wires, state of the oracle simulators, retained coins and outcomes, previous versions of saved outputs and state changes and feature workspace. The extra term covers lattice-instance and interface inputs. External reference systems are not part of the algorithm's storage. These totals are obtained by composing bounded operations and charging both computation and uncomputation. Parsers and oracle calls return with their temporary registers cleared. Retained histories contain the declared outputs and state changes, without copying the adversary's quantum registers.

An explicit forgery bound. Let $L = \Delta_F + \Delta_\mu + P_{\mathrm{pair}}$, let $\mathcal{B}$ denote the complete right-hand side of the theorem's final inequality, and set $a = 2^{-255}$. Since $J \geq 1$, both finite-selection probabilities are positive. The bound implies

$$\epsilon \leq \min\left\{1,\ L + \frac{(2q_B + 1)^2}{\lambda_{MR}} \left(a + \left((1 - a)\mathcal{B}\right)^{1/3}\right)\right\}.$$

Indeed, for $x \geq a$ one has $(x - a)^3 \leq x(x - a)^2$. For $x < a$ the resulting upper bound on x is immediate. Substitute $x = \lambda_{MR}[\epsilon - L]_+/(2q_B + 1)^2$ to obtain the claim. The target-selection factor K_* remains inside $\mathcal{B}$ and, for strong unforgeability, inside P_{pair}.

The theorem's oracle simulation requires bounded query input and output lengths, with every encoded length representable in 64 bits. One may take input width $\max\{I_{\mathrm{adv}}, 8192 + C_{\max} + M_{\max}\}$ bytes, since every honest Pretzel-(5,5) hash input has at most 8192 bytes apart from the context and message fields, and output width $\max\{L_{\mathrm{adv}}, 739{,}508{,}224\}$ bytes, padded to whole 13-bit words as for Pretzel-(4,4). The experiment observes the matrix through the functional interface; it does not observe the running time of the implementation's rejection sampling or how far it reads into private streams. The costs of the finite matrix oracle simulator and of the adapter for complete inputs must be charged as for Pretzel-(4,4).

Bounded setup and encoding. Apply the bound to the 1908-byte encoding of Section I.1. The work bounds above cover finite sampling distributions, access to private randomness, advice, and finite oracle simulation. The functional hypotheses use the bounded setup without tests of Section I.1, the canonical encoding, and the mask sampler comparison.

Each key-generation call uses zero public test samples. Counting both evaluation and uncomputation, the setup component count is

$$2(512 + 2048 + 8 + 2 + 2)K = 5144K \leq 7448K.$$

The terms count key candidates, dither candidates, matrix hash inputs, integrity hashes and expansions of the selected dither. The independent key search for loop control (Appendix B.5) uses internal coins, not

additional key-seed oracle queries. Its work is included in C_{setup}. The theorem uses the allowance $7448K$; all signing, verification, matrix, challenge and proof-only public response-sampling charges, padded widths and execution multipliers are as stated. Setup generates no public sample rows. The public response sampler's $e_{\text{pub},j}$ term belongs to the proof's response sampling, not to setup.

The encoding is injective and its public caps only restrict the accepted response set. Proposition D.4 preserves the comparison of all outputs and state kept for later calls, and the extraction and secant bounds, without an encoding error term. The common public setup continuation adds no term for selecting a candidate. Its availability bound is d, as proved in Section I.1.

The extraction projectors use the folded-index construction of Appendix C.2, with the unused raw bit retained in the workspace. The average remains over uniform 256-bit raw programmed values, so the displayed $f_{2^{255}}$ and all constants are unchanged.

Encoding and private randomness. Proposition D.4 preserves the comparison of all outputs and state kept for later calls under public encoding. The same total bounded dither selection of Section I.1 is applied on both sides, including failure. Appendix E therefore gives the key-replacement factor $\eta_{55}(1-\eta_{55}^{512})$.

The two independent private rows of 8544 bytes are rearranged by a fixed bijection into a proposal tape of 10,944 bytes and a transport tape of 6144 bytes. Thus the transport tape is independent of the proposal at the point in the proof where private hash families have been replaced. Each step reuses its six random limbs for the same decisions as before. This argument does not assume independence of the implementation's SHAKE streams before that replacement. Checks that the response fits in the encoding, label recovery and encoding read only responses and public data. They introduce no new reader of the private PRF keys. The advice remains independent of fresh experiment randomness and is neither copied nor reflected about.

One format per key. Fix one format P throughout the experiment, with one format per key, as required in Section I.1. On the existing event G that PRF keys are distinct, and after the private-family replacement removes the external overlay at cost e_P, an internal row $R[i,r,u]$ can be renamed $R[P,i,r,u]$. Adding and removing fixed public P are inverse reindexings. They preserve roles, offsets and all saved attempt outputs and state changes used when an attempt name repeats. The external query interface is unchanged. The public encoding map is then applied at the same point in the comparison of real and simulated signing, where the baseline is independent of the PRF keys. The existing e_P and δ_G terms are each charged once. No new error term or hardness assumption is introduced by this relabeling. It needs no separate format field in the private hash inputs.

Matrix representation. The NTT is a fixed bijection because each butterfly has determinant $-2\zeta \neq 0$ over $\mathbb{F}_{7681}$. For the auxiliary instance, first take the coefficient-ring adjoint and split it as in Appendix B.5, then convert to NTT coordinates. These bijections preserve uniformity and the independence required by the finite oracle simulator, without changing the secret or error distribution. The matrix and raw streams retain their required joint law, with the same δ_M and $2\delta_M$ charges and coefficients $7+2\kappa_{55}$ and 3 in the theorem.

I.3 Matched Pretzel-(5,5) attack estimates

Table 10 compares estimated attack costs for Pretzel-(5,5) and ML-DSA-87. Each cell is the minimum among the generic attack costs returned by a fixed Lattice Estimator revision [1], under a cost model selected before optimization, with the dual-hybrid attack minimized as for Table 5. ADPS16 [2] and MATZOV [32] are classical models of lattice-reduction cost; ADPS16 quantum and ChaLoy21 [4] are quantum ones. For the quantum columns, the estimator reoptimizes each attack with the quantum costs; these columns do not cover every quantum algorithm. At the pinned revision, the ADPS16 classical, ADPS16 quantum, and

Role	Instance	ADPS16	MATZOV	ADPS16 quantum	ChaLoy21
Key recovery	Pretzel key	254.1	273.5	232.9	227.1
	ML-DSA-87 key	244.9	266.9	223.7	217.2
Extra assumption	Pretzel auxiliary MLWE	297.6	315.8	272.9	265.4
in the quantum proof	ML-DSA-87 self-target relaxation	265.4	288.2	240.9	233.6
Forgery	Pretzel SIS	358.0	374.3	324.9	315.1
	ML-DSA-87 strong MSIS	253.5	276.1	230.0	223.1

Table 10: Base-two logarithms of modeled operation counts. Each cell is the minimum returned cost for its instance and reduction-cost model. These values are attack estimates, not security lower bounds.

ChaLoy21 models charge $2^{0.292\beta}$, $2^{0.265\beta}$, and $2^{0.257\beta}$, respectively, for BKZ with block size β. These are the costs of the reduction subroutine supplied to the estimator. A full attack may add guessing, repetitions, or other work, so its total cost need not be this single exponential.

Under the two classical cost models, Pretzel-(5,5)'s estimated margin over ML-DSA-87 is much larger for the Module-SIS instance used for strong unforgeability than for key recovery: approximately 98–105 bits versus 6.7–10.0 bits. The two instances face different parameter constraints. Pretzel's small responses permit tight Euclidean extraction bounds, whereas key recovery depends on the dimensions, modulus, and secret and error distributions of the public-key relation. The larger margin concerns only the SIS term of the reduction, so it does not imply a corresponding gain in overall forgery security.

Among the three Pretzel instances, key recovery has the lowest estimated cost in all four models. The comparison supports ML-DSA-87 as Pretzel's design target. Generic LWE and SIS estimates do not cover every attack that exploits ring or module structure. Pretzel-(5,5) keeps the same 2^{255}-element challenge family, so the generic message search of Section 6 still takes order $2^{127.5}$ ideal quantum oracle evaluations; larger lattice dimensions do not increase this cost.

The Pretzel key instance has dimension and sample count $(1280, 1280)$ over modulus 7681, with CBD_1 secret and error. The auxiliary instance has dimension and sample count $(1280, 1536)$, with CBD_2 secret and CBD_6 error. The homogeneous SIS instance from extraction has $(n, m) = (1280, 2816)$ and squared Euclidean bound 58,848,400. This exceeds the parameter set's bound 58,685,600, so the estimate conservatively uses a larger solution radius. The key estimate uses unrestricted CBD_1 keys; transferring it to keys that satisfy the key-norm restriction is a separate step of the reduction.

The matched ML-DSA-87 key instance uses modulus 8,380,417 and uniform secret and error coefficients in $[-2, 2]$, with dimension and sample count $(1792, 2048)$. Its SIS comparisons use $(n, m, B_\infty) = (2048, 4096, 769{,}537)$ for the self-target relaxation and $(2048, 3840, 1{,}048{,}336)$ for the strong-unforgeability Module-SIS instance (strong MSIS in the table). The self-target relaxation drops the hash constraint and estimates a homogeneous SIS problem. Its bound is in the infinity norm, whereas Pretzel's extraction bound is Euclidean.

I.4 Pretzel-(5,5) measurements against ML-DSA-87

Compared with ML-DSA-87, Pretzel-(5,5) reduces the signature size from 4627 to 1908 bytes and also has smaller public and secret keys. Table 11 reports mean thread CPU times, described below it. The measured signer includes protections against timing leaks and erases the buffers that hold secret data. Each comparison runs on macOS (Apple M2 Pro) with messages of 32 bytes. AWS-LC uses a generic build, compiled without tuning for the processor.

Signing includes every retry and uses a prepared secret key. Verification with a key already unpacked

Operation	Pretzel	ML-DSA-87	Time ratio (95% interval)
Sign, prepared secret key (μs)	65.863	69.403	0.9491 [0.9427, 0.9569]
Verify prepared (μs)	5.893	7.025	0.8388 [0.8328, 0.8434]
Verify with preparation (μs)	23.809	49.792	0.4781 [0.4768, 0.4792]
Prepare public key (μs)	17.847	43.236	0.4128 [0.4110, 0.4145]
Generate key (μs)	54.56	55.14	0.9892 [0.9839, 0.9946]

Table 11: Mean thread CPU times on macOS (Apple M2 Pro virtual machines; absolute times are indicative), 32-byte messages; each row is a matched comparison. Verify prepared checks a signature with a public key that has already been prepared, that is, unpacked into the matrix and offset used in verification; Prepare public key is that unpacking; Verify with preparation does both, starting from the stored key.

excludes preparation. Verification starting from a stored key includes preparation of the public key.

We report mean thread CPU times, as at levels 2 and 3. Tables 1 and 3 use the ratios from mean thread CPU times.

Key generation. The measured Pretzel-(5,5) implementation generates a key in about 0.055 ms, against 0.055 ms for ML-DSA-87 (last row of Table 11), so it takes about as long; it runs once per key.