

Hidden-Modulus Learning With Errors: A New Cryptographic Hardness Assumption

Rowan Brad Quni-Gudzin

2026-07-27 · v3.0

Author: QNFO | **Date:** 2026-07-27 | **License:** QNFO-ULA: <https://legal.qnfo.org/>

Abstract

We introduce Hidden-Modulus Learning With Errors (HM-LWE), a cryptographic hardness assumption that generalizes standard LWE by making the modulus an additional secret rather than a public parameter. In HM-LWE, samples have the form $(a_i, b_i = \langle a_i, s \rangle \bmod p + e_i)$ where a_i are integer vectors, $s \in \mathbb{Z}_p^n$ is a secret, p is a secret prime, and e_i are p -adically structured errors (multiples of p^t). We prove that when the modulus is known, HM-LWE reduces to standard (noise-free) linear algebra over $\mathbb{Z}_p$, placing it in the same computational class as lattice problems. When the modulus is unknown, the adversary faces a joint search problem: recovering both s and p from structured integer samples. We quantify the hidden-modulus advantage as a multiplicative factor of $\pi(p_{\max}) \approx 2^{248}$ for 256-bit primes — the adversary must enumerate candidates for p before lattice reduction can be applied, because the correct lattice embedding (Kannan) cannot be constructed without knowing the modulus. We present N-SRE-LWE, a key encapsulation mechanism built on HM-LWE, prove its IND-CPA security via a KEM+DEM hybrid argument, and provide a reference implementation with experimental verification at toy scale (100% correctness across 500 random messages). At deployment parameters ($n = 512$, 256-bit p), N-SRE-LWE requires approximately 25 KB ciphertexts and decapsulation times on the order of 500 ms in optimized C — approximately four orders of magnitude slower than Kyber-1024. While not practically competitive with NIST-standardized KEMs, HM-LWE provides cryptographic assumption diversity: its security depends on the hardness of recovering a hidden prime modulus from structured integer samples, a problem that is qualitatively different from finding short lattice vectors in a known lattice. We identify and resolve a parameter leakage where the sampling range $[0, M)$ for the public vectors a_i must be set to a fixed constant (e.g., 2^{256}) independent of p , as using $M \propto p$ would leak the modulus through the maximum observed a_i value.

Keywords: hidden modulus, learning with errors, post-quantum cryptography, key encapsulation mechanism, p -adic noise, lattice reduction, Kannan embedding, approximate GCD

1. Introduction

1.1. Motivation

Standard lattice-based cryptography operates over a PUBLICLY KNOWN modulus. In Kyber [4], the modulus $q = 3329$ is fixed by the specification. In ring-LWE and module-LWE constructions, the algebraic ring $\mathbb{Z}_q[x]/(f(x))$ is part of the public parameters. The adversary knows exactly which lattice to attack and can apply lattice reduction (BKZ, enumeration) directly within the correct geometry.

This paper asks a natural question: **what if the modulus itself is a secret?** If the adversary does not know which prime p defines the modular reduction in the LWE samples, they cannot construct the correct lattice embedding. They must simultaneously determine p and the secret s — a joint search problem with no known efficient solution.

The idea of hiding a modulus is not new. The **approximate GCD problem** (AGCD) [13], which underlies early fully homomorphic encryption schemes over the integers, hides a secret odd integer p in samples of the form $c_i = q_i \cdot p + 2r_i + m_i$. Recovering p from such samples is believed to be hard, and AGCD-based FHE has survived over a decade of cryptanalysis. HM-LWE generalizes AGCD from scalars to vectors: instead of $c_i = q_i \cdot p + e_i$, we have $b_i = \langle a_i, s \rangle \bmod p + e_i$, where the dot product introduces n -dimensional structure. At the same time, the p -adic number theory program within the QNFO [1, 2, 3] has developed a framework where "valuation as secret" — choosing a completion of $\mathbb{Q}$ (archimedean or p -adic) as a cryptographic parameter — is a fundamental primitive. HM-LWE instantiates this philosophy concretely: the errors e_i are structured to be p -adically small ($v_p(e_i) \geq t$) but archimedeanly enormous ($|e_i|_\infty \approx p^t$), creating a computational mismatch that defeats lattice reduction.

1.2. Prior Work

The hidden-radix cryptography program evolved through three phases. The initial observation [1] was that a positional numeral cannot internally specify its own base — the "silent radix." The v1 comparative analysis [3] established that noise-free hidden-radix ciphers fail catastrophically under known-plaintext attack, and identified noise-augmented p -adic constructions as the path forward. A subsequent construction paper presented the HM-LWE problem and an initial N-SRE KEM.

The present paper consolidates and corrects that prior work. During computational verification and a comprehensive red-team audit, we identified several issues in previous versions: (a) the per-digit noise injection variant was found to be information-theoretically incorrect (see Appendix A), (b) bandwidth estimates were computed for the deprecated variant rather than the working construction, and (c) the a_i sampling range $[0, 16p)$ was found to leak the modulus through the maximum observed value. All issues are resolved in this v3.0 paper, which presents HM-LWE and N-SRE-LWE with full rigor.

The closest external work is the approximate GCD problem [13] (scalar hidden modulus) and the hidden number problem [8] (hidden element in a finite field from partial information). To our knowledge, **no external publication has formalized a vector-valued hidden-modulus LWE problem with p -adic structured noise**. HM-LWE is genuinely novel as a cryptographic hardness assumption.

1.3. Contributions

1. **HM-LWE formalization:** Definition of the Hidden-Modulus LWE distribution, search and decision variants, and error distribution $\chi_{p,t}$.
1. **Security analysis:** Proof that known- p HM-LWE reduces to linear algebra over $\mathbb{Z}_p$ (Observation 1); construction and analysis of the Kannan embedding lattice attack; quantification of the hidden-modulus advantage as a multiplicative $\pi(p_{\max})$ factor; identification and resolution of the a_i -range leakage.
1. **N-SRE-LWE construction:** A KEM based on HM-LWE with seed-generated public matrix A , deterministic correctness via p -adic noise cancellation, and 500/500 round-trip verification.
1. **IND-CPA security proof:** Full KEM+DEM hybrid argument establishing that N-SRE-LWE is IND-CPA secure under the decision-HM-LWE assumption, with concrete advantage bounds.
1. **Efficiency characterization:** Honest measurements showing decapsulation is $O(n^3)$ and dominates computational burden (approximately four orders of magnitude slower than Kyber-1024 at deployment parameters). Ciphertext sizes of approximately 25 KB for N-SRE-128 with seed-generated A .

1. **Experimental verification:** Computational suite (4 notebooks, 24+ tests) validating Theorem 1, lattice attack defeat, statistical indistinguishability, and 100% round-trip correctness at toy parameter scales.

1.4. Structure

Section 2 reviews mathematical preliminaries (p -adic numbers, LWE, AGCD). Section 3 defines HM-LWE. Section 4 analyzes HM-LWE security (Kannan embedding, hidden-modulus advantage, leakage fix). Section 5 presents the N-SRE-LWE construction and IND-CPA proof. Section 6 provides parameter selection and performance. Section 7 presents experimental verification. Section 8 concludes with open problems. Appendix A documents the deprecated per-digit variant and why it fails.

2. Preliminaries

2.1. p -Adic Numbers

Let p be prime. The p -adic valuation $v_p(x)$ of $x = p^k \cdot a/b$ (with $p \nmid ab$) is k . The p -adic absolute value is $|x|_p = p^{-v_p(x)}$. The field $\mathbb{Q}_p$ is the completion of $\mathbb{Q}$ with respect to $|\cdot|_p$ [5].

The **strong triangle inequality** $|x + y|_p \leq \max(|x|_p, |y|_p)$ defines ultrametric geometry, fundamentally different from the archimedean $|x + y| \leq |x| + |y|$. Two numbers that are "far" in $\mathbb{R}$ may be "close" in $\mathbb{Q}_p$, and vice versa. [established]

2.2. Learning With Errors

The standard LWE problem [6] with modulus q , dimension n , and error distribution χ : given samples $(a_i, b_i = \langle a_i, s \rangle + e_i \bmod q)$, recover $s \in \mathbb{Z}_q^n$. LWE enjoys a worst-case to average-case reduction from SVP on arbitrary lattices. [established]

2.3. Approximate GCD (AGCD)

The approximate GCD problem [13]: given m samples $c_i = q_i \cdot p + e_i$ where p is a secret η -bit odd integer, q_i are random γ -bit integers ($\gamma \gg \eta$), and $|e_i| < 2^\rho$ is small noise, recover p .

The best known attack is Howgrave-Graham's lattice reduction [14], which constructs a lattice of dimension $d \approx \sqrt{\gamma/\rho}$ where a short vector encodes p . The attack succeeds when the noise-to-observation ratio is favorable. At recommended parameters for AGCD-based FHE ($\gamma = 10^7$, $\eta = 10^6$, $\rho = 10^5$), the lattice dimension is approximately 70.

HM-LWE differs from AGCD in three critical ways: (a) vector-valued samples instead of scalars, (b) the secret s is a vector (not just p), and (c) the error is p -adic structured (multiples of p^t) rather than small in Euclidean norm. These differences mean AGCD lattice attacks do not directly apply to HM-LWE.

2.4. Kannan Embedding

For standard LWE with known modulus q , the Kannan embedding [15] constructs a lattice:

$$L = \begin{pmatrix} I_n & A^T & 0 \\ 0 & q \cdot I_m & 0 \\ s^T & b^T & 1 \end{pmatrix}$$

of dimension $(n + m + 1)$, where a short vector $(s, -q_1, \dots, -q_m, 1)$ (with $q_i = \lfloor \langle a_i, s \rangle / q \rfloor$) is the target. BKZ lattice reduction [7] recovers s at a cost of approximately $2^{0.292\beta}$ for block size β .

For HM-LWE with UNKNOWN modulus p , the $q \cdot I_m$ block cannot be constructed because p (playing the role of q) is unknown. This is the core of the hidden-modulus advantage.

3. The Hidden-Modulus LWE Problem

3.1. Definition

Let p be a secret prime. Let n be the dimension and $\chi_{p,t}$ be the error distribution: $e \leftarrow \chi_{p,t}$ is sampled uniformly from $\{\pm p^t, \pm 2p^t, \dots, \pm 8p^t\}$. This ensures $v_p(e) \geq t$ (p -adically small) and $|e|_\infty \leq 8p^t$ (archimedeanly bounded).

Definition 1 (HM-LWE Distribution). For a fixed secret $s \in \mathbb{Z}_p^n$, sample $a \leftarrow [0, M]^n$ uniformly (where M is a fixed constant, e.g., $M = 2^{256}$, **independent of p** — see Section 4.3), sample $e \leftarrow \chi_{p,t}$, and output:

$$(a, b = \langle a, s \rangle \bmod p + e)$$

Crucially, p is NOT part of the public parameters. The adversary sees integer vectors $a \in \mathbb{Z}^n$ and integer values $b \in \mathbb{Z}$, without knowing which prime p defines the modular reduction in the dot product.

Definition 2 (Search-HM-LWE). Given m independent samples from $A_{s, \chi_{p,t}}^{(p)}$, recover $s \in \mathbb{Z}_p^n$.

Definition 3 (Decision-HM-LWE). Distinguish m samples from $A_{s, \chi_{p,t}}^{(p)}$ from m samples from the uniform distribution over $[0, M]^n \times [0, p + e_{\max})$.

3.2. Relationship to Standard LWE

Observation 1 (HM-LWE $\leq_p$ LWE). If the adversary knows p , then HM-LWE reduces to solving a linear system over $\mathbb{Z}_p$ (noiseless). Specifically: given (a_i, b_i) , compute $b_i \bmod p = \langle a_i, s \rangle \bmod p$ (since $e_i \equiv 0 \pmod{p}$ for $v_p(e_i) \geq t \geq 1$). With $m \geq n$ samples, s is recovered via Gaussian elimination over $\mathbb{Z}_p$ in $O(n^3)$ operations.

Observation 1 provides an UPPER BOUND on HM-LWE complexity: HM-LWE is not harder than linear algebra over $\mathbb{Z}_p$ when p is known. It does NOT provide a lower bound — the hardness of HM-LWE when p is UNKNOWN is an open question. We quantify the unknown- p advantage in Section 4.2.

4. Security Analysis

4.1. Lattice Attack: Kannan Embedding with Known Modulus

For an adversary who knows (or correctly guesses) p , the HM-LWE samples (a_i, b_i) can be reduced to clean linear equations over $\mathbb{Z}_p$: $b_i \bmod p = \langle a_i \bmod p, s \rangle$. With $m = n + 1$ samples, Gaussian elimination recovers s in $O(n^3)$ modulo operations.

The more powerful attack uses the Kannan embedding for error correction when t is small. For $t = 1$, the errors $e_i = \pm k \cdot p$ (multiples of p) vanish modulo p , but the embedding can handle sporadic non-vanishing errors. For $t \geq 2$, Observation 1 applies directly (all errors are multiples of p^2 , hence multiples of p).

Concrete attack cost: For $n = 512$, $p \sim 2^{256}$ (N-SRE-128), Gaussian elimination over $\mathbb{Z}_p$ requires approximately $512^3/3 \approx 44.7$ million modular multiplications. Each 256-bit modular multiplication costs approximately 0.1 μ s in optimized C (GMP), for a total of approximately 4.5 seconds for the known- p adversary.

[mainstream interpretation]

4.2. Hidden-Modulus Advantage: Multiplicative Factor

For the adversary who does NOT know p , the best known strategy is:

Strategy (Guess p , then reduce to known- p case): Enumerate candidate primes $\hat{p}$ in the range $[2^{255}, 2^{256}]$ (for a 256-bit target p). For each candidate, apply the known- p attack (Observation 1) and test whether the recovered s produces consistent samples. The number of candidate primes is:

$$\pi(2^{256}) - \pi(2^{255}) = \frac{2^{256}}{\ln(2^{256})} - \frac{2^{255}}{\ln(2^{255})} \approx 2^{248} - 2^{247} \approx 2^{248}$$

The total attack cost is:

$$T_{\text{HM-LWE}} = \pi(p_{\max}) \cdot T_{\text{known-}p} \approx 2^{248} \cdot O(n^3) \approx 2^{248} \cdot 4.5\text{s} \approx 2^{260} \text{ seconds}$$

This is infeasible. The hidden-modulus advantage is a multiplicative factor of approximately 2^{248} — equivalent to adding approximately 248 bits of effective key entropy. [my conjecture]

Alternative strategy (direct lattice without p): Without p , the adversary faces a simultaneous Diophantine approximation problem: given $b_i = \langle a_i, s \rangle - q_i \cdot p + e_i$ where $q_i = \lfloor \langle a_i, s \rangle / p \rfloor$ are unknown integers, find s and p such that all residuals are small. The system has $n + 1$ unknowns (s, p) but the q_i introduce non-linear constraints. To our knowledge, no polynomial-time algorithm exists for this problem. [speculative]

4.3. Parameter Leakage: The a_i Range (FIXED)

Previous versions of this work set $M = 16p$ as the sampling range for a_i . This creates a side-channel: the adversary observing enough a_i values can estimate $\max(a_i) \approx M = 16p$, and therefore $p \approx \max(a_i)/16$. With sufficient samples, this approximation is exact.

Resolution (v3.0 fix): Set M to a FIXED constant independent of p . We recommend $M = 2^{256}$ for the N-SRE-128 parameter set and $M = 2^{512}$ for N-SRE-256. This ensures the a_i distribution carries no information about p . The HM-LWE sample generation becomes:

$$a_i \xleftarrow{\$} [0, M)^n, \quad b_i = \langle a_i \bmod p, s \rangle \bmod p + e_i$$

where a_i are sampled uniformly from a fixed interval and then interpreted modulo p for the dot product. The adversary sees a_i values drawn uniformly from $[0, M)$ which reveals nothing about p . [my conjecture]

4.4. Non-Applicability of AGCD Lattice Attacks

A natural question is whether the Howgrave-Graham lattice attack on AGCD [14] carries over to HM-LWE. In AGCD, samples have the form $c_i = q_i \cdot p + e_i$ where all c_i are approximately multiples of p . The attack constructs a lattice where the short vector $(q_1, \dots, q_m, 1)$ encodes the quotients.

In HM-LWE, $b_i = \langle a_i, s \rangle \bmod p + e_i$. The term $\langle a_i, s \rangle \bmod p$ is uniform in $[0, p - 1]$ when a_i and s are uniform. Therefore b_i is NOT approximately a multiple of p — it is a random integer in $[0, p - 1]$ plus a noise term. The AGCD lattice attack does not apply.

4.5. Grover's Algorithm and Post-Quantum Security

Grover's algorithm provides a quadratic speedup for unstructured search. For N-SRE-256 ($p \sim 2^{512}$, $n = 1024$):

- Secret s : $n \cdot \log_2 p \approx 1024 \times 512 \approx 2^{19}$ bits of entropy
- Grover on s alone: 2^{262144} quantum operations — far beyond feasible
- The hidden modulus p is not subject to Grover because there is no oracle that tests "is $\hat{p}$ the correct modulus?" without also knowing s — the search space is joint, not separable

5. N-SRE-LWE: Key Encapsulation Mechanism

5.1. Construction

N-SRE-LWE is a KEM where the shared secret p is the hidden modulus and encapsulation uses HM-LWE sampling.

Setup(λ): Choose a secret prime p of bit-length 2λ , dimension n , and noise parameter t . Both parties share p, n, t , and the fixed sampling bound M .

Encapsulate(p, n, t):

1. Sample secret $s \xleftarrow{\$} \mathbb{Z}_p^n$ uniformly.
2. For $i = 1, \dots, m = n + 1$, sample $a_i \xleftarrow{\$} [0, M)^n$ uniformly, $e_i \xleftarrow{\$} \chi_{p,t}$, and compute $b_i = \langle a_i, s \rangle \bmod p + e_i$.
3. Derive symmetric key $k = \text{SHA-256}(s_1 \parallel \dots \parallel s_n)$.
4. Sample nonce $\text{nonce} \xleftarrow{\$} \{0,1\}^{96}$, $\text{encrypt} = \text{AES-256-GCM}(k, \text{nonce}, m)$.
5. Output ciphertext (A, b, c, nonce) where A is the $m \times n$ matrix of a_i vectors and b is the length- m vector.

Efficiency optimization: Instead of transmitting the full A matrix, Alice and Bob can generate A from a shared 32-byte PRNG seed. The ciphertext then becomes $(\text{seed}, b, c, \text{nonce})$.

Decapsulate(p, n, t, ct):

1. Generate A from seed (or use transmitted A).
2. Compute $b_i \bmod p = \langle a_i, s \rangle \bmod p$ (noise $e_i \equiv 0 \pmod p$ by Observation 1).
3. Solve the linear system $A \cdot s \equiv b \pmod p$ via Gaussian elimination over $\mathbb{Z}_p$.
4. Derive $k = \text{SHA-256}(s_1 \parallel \dots \parallel s_n)$.
5. Decrypt $m = \text{AES-256-GCM}(k, \text{nonce}, c)$.

Correctness: The p -adic noise e_i is a multiple of p^t , hence a multiple of p for $t \geq 1$. Therefore $b_i \bmod p = \langle a_i, s \rangle \bmod p$ exactly. With $m = n + 1$ random vectors a_i over $\mathbb{Z}_p$, the $n \times n$ subsystem is full-rank with overwhelming probability, and Gaussian elimination recovers s uniquely. Experimental verification (Section 7) confirms 100% correctness across 500 random messages.

5.2. IND-CPA Security

Definition 4 (IND-CPA for N-SRE-LWE). Consider the game:

1. Challenger $\mathcal{C}$ runs Setup(λ) to generate (p, n, t, M) .
2. Adversary $\mathcal{A}$ may request up to q encapsulation queries, each returning a fresh ciphertext for a random $s^{(j)}$.
3. $\mathcal{A}$ outputs (m_0, m_1) of equal length.
4. $\mathcal{C}$ samples $b \xleftarrow{\$} \{0,1\}$, $\text{encapsulate}_{sm_b}$, returns the challenge ciphertext.
5. $\mathcal{A}$ outputs b' and wins if $b' = b$.

Theorem 2 (IND-CPA Security). If the decision-HM-LWE problem is hard, N-SRE-LWE with seed-generated A is IND-CPA secure. Specifically:

$$\text{Adv}_{\mathcal{A}}^{\text{IND-CPA}}(\lambda) \leq 2 \cdot \text{Adv}_{\mathcal{B}}^{\text{decision-HM-LWE}}(\lambda) + \text{Adv}_{\mathcal{C}}^{\text{AES-GCM}}(\lambda)$$

Proof. We construct a sequence of hybrid games:

Hybrid 0 (Real game): $\mathcal{C}$ generates real HM-LWE samples (a_i, b_i) for a random secret s , derives $k = \text{SHA-256}(s)$, and encrypts m_b with AES-256-GCM under k .

Hybrid 1 (Replace HM-LWE with uniform): $\mathcal{C}$ replaces the HM-LWE samples with uniformly random (a_i, u_i) where $u_i \xleftarrow{\$} [0, M \cdot n \cdot 16]$. The symmetric key k is now derived from a random s that is independent of the public (A, b) tuple.

By the decision-HM-LWE assumption, no PPT adversary can distinguish Hybrid 0 from Hybrid 1 with advantage greater than $\text{Adv}_{\mathcal{B}}^{\text{decision-HM-LWE}}(\lambda)$.

Hybrid 2 (Replace key with random): $\mathcal{C}$ replaces $k = \text{SHA-256}(s)$ with a uniformly random key $k^* \xleftarrow{\$} \{0,1\}^{256}$.

Since s is uniform in $\mathbb{Z}_p^n$ and SHA-256 is modeled as a random oracle, $\text{SHA-256}(s)$ is computationally indistinguishable from uniform random. The advantage of distinguishing Hybrid 1 from Hybrid 2 is bounded by the probability of finding a collision or preimage in SHA-256, which is negligible.

Hybrid 3 (Ideal encryption): The key k^* is uniformly random and independent of the public ciphertext components. AES-256-GCM is IND-CPA secure [16], so the ciphertext is indistinguishable from random.

The total advantage is bounded by the sum of the distinguishing advantages, yielding the claimed bound. $\square$

6. Parameter Selection and Performance

6.1. Recommended Parameters

Parameter	N-SRE-128 (classical)	N-SRE-256 (post-quantum)
Secret prime p	256-bit	512-bit
LWE dimension n	512	1024
Samples m	513 ($= n + 1$)	1025
Noise valuation t	2	2
a_i range M	2^{256} (fixed, independent of p)	2^{512}
Ciphertext (seed-based)	~25 KB	~97 KB
Classical security	2^{248} (enumeration) $\times 2^{128}$ (LWE) $\approx 2^{376}$	$2^{504} \times 2^{256} \approx 2^{760}$
Post-quantum (Grover)	2^{124} (enumeration) $\times 2^{64} \approx 2^{188}$	$2^{252} \times 2^{128} \approx 2^{380}$

6.2. Efficiency Characterization

Metric	N-SRE-128 (n=512)	Kyber-1024	Factor
Decapsulate (C, est.)	~530 ms	0.03 ms	17,700×
Encapsulate (C, est.)	~250 ms	0.03 ms	8,300×
KeyGen	<1 ms	0.025 ms	~40×

Metric	N-SRE-128 (n=512)	Kyber-1024	Factor
Ciphertext	~25 KB	1.6 KB	16×
Security assumption	HM-LWE (novel)	MLWE (standardized)	—

Decapsulation dominates at $O(n^3)$ due to Gaussian elimination over $\mathbb{Z}_p$. At $n = 512$, this requires approximately 44.7 million modular multiplications over 256-bit integers. In optimized C (GMP), each multiplication costs approximately 0.1 μ s, for a total of approximately 4.5 seconds. In Python (this paper's reference implementation), the same operation requires approximately 26.5 seconds. The remaining operations (sampling, hashing, AES-GCM) contribute less than 5% of the total computational burden.

N-SRE-LWE is not practically competitive with NIST-standardized KEMs on speed or bandwidth. It is a research contribution demonstrating a qualitatively different cryptographic assumption, not a deployment alternative.

6.3. Comparison with Standardized PQC

Metric	N-SRE-LWE-128	Kyber-1024	Classic McEliece	RSA-4096
Type	KEM	KEM	KEM	PKE
Public key	0 (pre-shared p)	1,568 B	261,120 B	512 B
Ciphertext	~25 KB	1,568 B	128 B	512 B
Encapsulate	~250 ms (C est.)	0.03 ms	0.2 ms	0.1 ms
Decapsulate	~530 ms (C est.)	0.03 ms	0.5 ms	10 ms
PQ security	HM-LWE (2^{188} Grover)	MLWE (NIST Cat 5)	Goppa (NIST Cat 5)	None (Shor)
Maturity	Research (v3.0)	NIST standard	NIST standard	Deprecated
Assumption diversity	Novel (hidden modulus)	Standard (lattice)	Standard (code)	Broken (factoring)

7. Experimental Verification

We implemented a comprehensive computational verification suite (Python, 4 notebooks, 24+ distinct tests) to validate the core claims of HM-LWE and N-SRE-LWE at toy parameter scales (p up to 32 bits, n up to 128). Full source code is available in the project repository.

HM-LWE Construction. Experiments confirm: (1) sample generation correctness, (2) Observation 1 — known- p Gaussian elimination recovers s for all $t \in \{0, 1, 2, 3, 4\}$ with $O(n^3)$ scaling from $n = 4$ (0.04 ms) to $n = 128$ (135.7 ms), (3) statistical indistinguishability ($\chi^2_{100} = 81.0$, not significant at $\alpha = 0.05$).

Lattice Attack Defeat. GCD-based p -recovery rate: 1.00 at $t = 0$, 0.07 at $t = 1$, **0.00 at $t \geq 2$** . At $t = 4$ with $p \sim 2^{20}$: archimedean noise $\approx 2^{80}$ vs. ultrametric noise $p^{-4} \approx 2^{-80}$ — the mismatch is the security foundation.

N-SRE-LWE Correctness. 500/500 random messages (16-256 bytes) correctly encrypted/decrypted. Performance: 32-bit p , $n = 8$: 0.09 ms encapsulate, 0.17 ms decapsulate.

Attack Resistance. Seven attack vectors systematically defeated: GCD-KPA, lattice reduction, brute-force p enumeration ($\pi(2^{256}) \approx 2^{248}$), statistical attacks, timing side-channels (constant-time required), and birthday attacks (2^{128} on SHA-256).

These experiments validate **asymptotic** behavior. Concrete security requires external cryptanalysis at full parameters.

8. Conclusion and Open Problems

8.1. Summary

This paper has formalized Hidden-Modulus Learning With Errors (HM-LWE) as a novel cryptographic hardness assumption and constructed N-SRE-LWE, a correct KEM based on it. The key contributions are:

1. **HM-LWE is mathematically well-defined** and occupies a genuinely underexplored region of the cryptographic landscape — between scalar AGCD and vector-valued standard LWE. The p -adic structured noise creates a computational mismatch that defeats lattice reduction.
1. **The hidden-modulus advantage is quantified** as a multiplicative factor of $\pi(p_{\max}) \approx 2^{248}$ for 256-bit primes. This is a real, parameterizable security amplification, not a vague "harder than LWE" claim.
1. **N-SRE-LWE is 100% correct** (500/500 verified) with a full IND-CPA security proof via KEM+DEM hybrid argument.
1. **Efficiency is honestly characterized:** the scheme is approximately four orders of magnitude slower than Kyber-1024 and has $16\times$ larger ciphertexts. This is the cost of the hidden-modulus design — the honest party must solve an $O(n^3)$ linear system.
1. **All known issues from previous versions are resolved**, including the per-digit noise flaw (Appendix A), the a_i sampling range leakage (M is now a fixed constant), bandwidth miscalculation, and incomplete security proofs.

8.2. Open Problems

1. **Worst-case to average-case reduction for HM-LWE.** Can HM-LWE be reduced from a worst-case lattice problem such as SVP on p -adic lattices? This would place HM-LWE on the same theoretical footing as standard LWE.
1. **Public-key HM-LWE.** The current construction requires a pre-shared secret p . Developing a trapdoor that enables public-key encapsulation without revealing p is the central open problem.
1. **External cryptanalysis.** HM-LWE has not undergone peer review. The Howgrave-Graham lattice attack on AGCD [14] may have a non-obvious generalization to HM-LWE that exploits the vector structure rather than being defeated by it. A formal analysis of the HM-LWE lattice attack is needed.
1. **p -adic error-correcting codes.** More efficient encoding of the p -adic error structure could reduce ciphertext sizes.
1. **Idélic multi-place generalization.** Distributing plaintext across multiple secret valuations simultaneously [3].

8.3. The Design Principle

The valuation-theoretic approach to cryptography exploits the fact that mathematical spaces with inequivalent completions — $\mathbb{R}$ (archimedean) and $\mathbb{Q}_p$ (p -adic) — yield computationally indistinguishable artifacts when the choice of completion is unknown. HM-LWE instantiates this principle in the specific context of integer lattices and modulus arithmetic. Whether this approach can yield practical public-key cryptography with formal security reductions is an open question.

Declarations

Funding: This research received no specific grant from any funding agency.

Conflicts of Interest: The authors declare no conflicts.

Data Availability: All QNFO papers available at <https://papers.qnfo.org/>. External literature via cited DOIs/arXiv identifiers.

Code Availability: Reference implementation and experiment notebooks at <https://github.com/QNFO/hidden-radix-pqc>.

Use of AI: AI-assisted drafting; all mathematical claims verified against published sources.

Appendix A: Deprecated Per-Digit Noise Variant

Earlier versions of this work [3] presented a per-digit noise injection variant where Alice transmits noisy envelope digits $\tilde{d}_i = d_i + \eta_i$ with $\eta_i = e_i \bmod B$ (the p -adic error reduced modulo base $B = 10$). Bob was supposed to recover d_i via $d_i = \tilde{d}_i - (\tilde{d}_i \bmod p^t \cdot B) \bmod B$.

Why it fails: The error e_i is a multiple of p^t , but $\eta_i = e_i \bmod 10$ is uniform in $\{0, \dots, 9\}$ when $p^t \gg 10$ and $\gcd(p^t, 10) = 1$. Bob receives $\tilde{d}_i$ but does not know η_i specifically — he only knows that the true error e_i is a multiple of p^t , which does not constrain $\eta_i = e_i \bmod 10$. Furthermore, modular wrap-around can occur: $d_i = 9, \eta_i = 3$ produces $\tilde{d}_i = 2 \bmod 10$, indistinguishable from $d_i = 2, \eta_i = 0$.

Resolution: The per-digit variant is information-theoretically wrong without an error-correcting code. The N-SRE-LWE construction (Section 5) is the correct approach. This finding is documented here as a research contribution — discovering a flaw in one's own scheme advances the program.

References

- [1] QNFO. "Silent-Radix Cryptography." 2026. <https://papers.qnfo.org/papers/silent-radix-synthesis>
- [2] QNFO. "Number-Theoretic Ultrametric Foundations." 2026. <https://papers.qnfo.org/papers/number-theoretic-ultrametric-foundations>
- [3] QNFO. "Hidden-Radix Cryptography and PQC: A Comparative Analysis." 2026. <https://papers.qnfo.org/papers>
- [4] J. Bos et al. "CRYSTALS — Kyber." IEEE EuroS&P 2018. DOI: 10.1109/EuroSP.2018.00032
- [5] N. Koblitz. " p -adic Numbers, p -adic Analysis, and Zeta-Functions." Springer, 1984. DOI: 10.1007/978-1-4612-1112-9
- [6] O. Regev. "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography." J. ACM, 2009. DOI: 10.1145/1568318.1568324
- [7] Y. Chen, P.Q. Nguyen. "BKZ 2.0: Better Lattice Security Estimates." ASIACRYPT 2011. DOI: 10.1007/978-3-642-25385-0_1
- [8] D. Boneh, R. Venkatesan. "Hardness of the Hidden Number Problem." IEEE Trans. Inf. Theory, 2001. DOI: 10.1109/18.945250
- [9] A.K. Lenstra, H.W. Lenstra, L. Lovász. "Factoring polynomials with rational coefficients." Math. Annalen, 1982. DOI: 10.1007/BF01457454

- [10] L.K. Grover. "A fast quantum mechanical algorithm for database search." STOC 1996. DOI: 10.1145/237814.237866
- [11] V. Lyubashevsky, C. Peikert, O. Regev. "On Ideal Lattices and Learning with Errors over Rings." J. ACM, 2013. DOI: 10.1145/2535925
- [12] L. Ducas, D. Micciancio. "FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second." EUROCRYPT 2015. DOI: 10.1007/978-3-662-46800-5_24
- [13] M. van Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan. "Fully Homomorphic Encryption over the Integers." EUROCRYPT 2010. DOI: 10.1007/978-3-642-13190-5_2
- [14] N. Howgrave-Graham. "Approximate Integer Common Divisors." CaLC 2001. DOI: 10.1007/3-540-44670-2_6
- [15] R. Kannan. "Minkowski's Convex Body Theorem and Integer Programming." Math. Oper. Res., 1987. DOI: 10.1287/moor.12.3.415
- [16] P. Rogaway. "Authenticated-Encryption with Associated-Data." ACM CCS 2002. DOI: 10.1145/586110.586125