DTRU: A Versatile, Compact, Simple, and Robust NTRU KEM with Double E_8 Encoding

Hengchuan Zou[1], Songlin Li[1], Jieyu Zheng[1], Xiaowen Hu[1], Hanyu Wei[1], Weizhi Ao[1], Yifan Dong[1], Wenbo Guo[1], and Yunlei Zhao[1]

College of Computer Science and Artificial Intelligence, Fudan University, China
ylzhao@fudan.edu.cn

Abstract. Responding to China's 2025 call for commercial cryptographic standards mandating 128-bit, 256-bit, and 512-bit security (optional 384-bit), we propose DTRU, a versatile, compact, simple, and robust NTRU-based key encapsulation mechanism (KEM). Our principal design contribution is double E_8 encoding, which constructs 16-dimensional lattice codes from E_8 with low decoding complexity. We further provide a detailed analysis of decryption-failure probability under this encoding mechanism.

DTRU's design achieves a careful balance among versatility, compactness, simplicity, and robustness. To accommodate diverse application requirements, it supports multiple ring structures, including power-of-two cyclotomic rings, tricyclotomic rings, and large-Galois-group prime-degree prime-ideal number fields (LPPNF). The double E_8 encoding enables DTRU to achieve enhanced error correction with compact bandwidth. The design prioritizes simplicity to facilitate deployment on low-power devices, achieved by eschewing additional coefficient compression techniques and redundant invertibility checks during key generation, while enabling circuit/code reuse. Security robustness is guaranteed through parameter selections that offer adequate security redundancy, mitigating potential cyclotomic ring risks via LPPNF, and precluding sparse noise distributions in the recommended parameter sets.

Complementing our theoretical advances, we present comprehensive implementations of all the parameter sets with dedicated support for C, AVX2, and ARM platforms, leveraging architecture-specific optimizations. For example, compared to NTRU-HRSS and Kyber at the same security levels, our KEM is 49%-52% more compact and 3.84–15.69× faster than NTRU-HRSS in the round-trip time of ephemeral key exchange, and is 7%-27% more compact and 1.05–1.32× faster than Kyber.

Keywords: Lattice based Cryptography · NTRU · Key Encapsulation Mechanism

1 Introduction

Shor's quantum algorithm [35] can efficiently solve problems such as integer factorization, undermining the security of many widely used public-key schemes

and motivating the development of post-quantum cryptography (PQC). One prominent direction within PQC is based on NTRU lattices. NTRU [23], first proposed in 1998, has played a foundational role in many modern cryptographic constructions [13,17,19,14,6,28]. Contemporary NTRU-style schemes are typically justified under two hardness assumptions: the decisional NTRU assumption and the Ring-LWE assumption (some variants instead rely on the Ring-LWR assumption). During the NIST PQC standardization process NTRU reached the third round as a finalist but was not selected; a main reason was that it was neither the fastest nor the most compact among the lattice-based KEM finalists. Consequently, much subsequent work has concentrated on improving NTRU's efficiency and reducing its bandwidth.

The original NTRU's ring choice $\mathbb{Z}_q[x]/(x^n-1)$ with prime n is incompatible with the Number Theoretic Transform (NTT). Although alternative multiplication strategies (e.g., NTT-unfriendly methods [10] or Toom-Cook [3]) are available, they incur performance overheads. Moving to NTT-friendly rings is therefore a natural optimization to achieve substantial speedups, as in NTTRU [31] and NTRU+ [26]. Another line of work, exemplified by BAT [18], focuses on ciphertext compression by adopting the Ring-LWR assumption and exploiting NTRU structure for faster integer arithmetic, albeit at the cost of increased key-generation complexity.

Recent approaches like NEV [37] and DAWN [30] seek to optimize the bandwidth-performance tradeoff. NEV employs vector encoding that maps plaintext bits to multiple coefficient MSBs (similar to repetition coding), enhancing decryption robustness. Its variant NEV′, while improving noise tolerance, relies on the non-standard sspRLWE assumption. DAWN utilizes zero-divisor encoding for stronger ciphertext compression with negligible decryption failure, and introduces a double-encoding paradigm. However, DAWN requires an additional polynomial inversion and employs sparse noise. Both schemes are confined to power-of-two cyclotomic rings $\mathbb{Z}_q[x]/(x^n+1)$, restricting ring and parameter choices.

1.1 Our Contributions

In October 2025, the Institute of Commercial Cryptography Standards of China issued a call for commercial cryptographic algorithm standards, mandating support for at least three classical security strength levels: 128-bits, 256-bits, and 512-bits, with optional 384-bits support[1]. In response to this requirement, we present DTRU, a versatile, compact, simple, and robust NTRU-based key encapsulation mechanism that achieves provable IND-CCA security under the NTRU and Ring-LWE assumptions.

Versatility through Multiple Instantiations DTRU supports multiple mathematical instantiations to accommodate diverse security and performance requirements. Our primary instantiation utilizes tricyclotomic rings $(\mathbb{Z}_q[x]/(x^n-x^{n/2}+$

[1] https://www.niccs.org.cn/niccs/

1), following the approach in [31]), which enables seamless support for classical security levels spanning 128, 192, 256, 384, and 512 bits. For enhanced security assurance, we introduce DTRU-Prime, which operates on large-Galois-group prime-degree prime-ideal number fields (LPPNF, specifically $\mathbb{Z}_q[x]/(x^n - x - 1)$) to provide conservative defense against potential algebraic attacks through subring- and subfield-free structures. Additionally, we present DTRU-Light, a lightweight variant based on power-of-two cyclotomic rings ($\mathbb{Z}_q[x]/(x^n + 1)$) optimized for resource-constrained devices. This multi-faceted instantiation capability underscores the exceptional *versatility* of the DTRU framework.

Compactness via Advanced Encoding and Simple Design We employ *double* E_8 *encoding* to enhance error correction capabilities, achieving remarkable *compactness*. The design prioritizes *simplicity* to facilitate deployment on low-power devices, achieved by eschewing additional coefficient compression techniques and redundant invertibility checks during key generation, while enabling circuit/code reuse.

Robust Security Design DTRU's robust security design incorporates adequate parameter security redundancy, LPPNF instantiation ($\mathbb{Z}_q[x]/(x^n - x - 1)$ as in [7]) with subring- and subfield-free structures to defend against algebraic attacks, and the exclusion of sparse noise distributions through centered binomial distributions (parameter at least 2 for most configurations). These measures collectively enhance DTRU's security *robustness*.

Technical Innovation: Double E_8 *Encoding* Our key technical contribution is a novel lattice-encoding technique, *double* E_8 *encoding*, which synergistically combines NEV's repetition encoding with an extended Hamming-style code derived from the E_8 lattice. This approach achieves superior error correction by mapping 4 bits to 16 coefficients, resulting in a significantly enhanced minimum Euclidean distance that dramatically reduces decryption failure rates while maintaining efficient decoding performance.

Practical Efficiency and Performance Validation Our comprehensive implementation and benchmarking demonstrate DTRU's practical efficiency. In C implementations, DTRU achieves 8.76-15.69× speedup over traditional NTRU schemes and 1.05-1.32× improvement over Kyber. With AVX2 optimizations, DTRU maintains 3.84-6.28× performance gains over NTRU schemes while delivering 1.10-1.22× improvement over Kyber. On ARM Cortex-M4 embedded systems, DTRU consistently outperforms Kyber with 1.06-1.62× speedup. These results validate the practical efficiency of our double E_8 encoding approach across diverse computing platforms.

1.2 Comparison to the State of the Art

Table 1 compares DTRU with DAWN [30], NTRU-HPS [23], NTRU-HRSS, SNTRU-Prime [7], NEV [37] and Kyber [34]. Here, CT and PK denote ciphertext

Table 1: Comparison between DTRU, NTRU KEMs and Kyber in Size

Scheme	CT (bytes)	PK (bytes)	Improve Ratio	Error rate	Concrete Security	Security Level
DTRU-512	640	768	-	2^{-134}	150	128 bits
DTRU-Light	512	640	-	2^{-144}	151	
DAWN-α-512	436	615	-9%	2^{-133}	140	
NTRU-HPS$^{677}_{2048}$	930	930	38%	$2^{-\infty}$	165	
NTRU-HRSS-701	1138	1138	49%	$2^{-\infty}$	156	
NEV-512	615	615	6%	2^{-138}	137	
Kyber-512	768	800	27%	2^{-138}	140	
DTRU-768	960	1152	-	2^{-186}	213	192 bits
NTRU-HPS$^{821}_{4096}$	1230	1230	14%	$2^{-\infty}$	193	
Kyber-768	1088	1184	7%	2^{-164}	196	
DTRU-1024	1280	1536	-	2^{-190}	280	256 bits
NTRU-HPS$^{1229}_{4096}$	1842	1842	37%	$2^{-\infty}$	282	
NTRU-HRSS-1373	2401	2401	52%	$2^{-\infty}$	289	
Kyber-1024	1568	1568	27%	2^{-174}	262	
DTRU-1024$'$	1024	1280	-	2^{-212}	276	
DAWN-α-1024	973	1229	-5%	2^{-163}	270	
NEV-1024	1229	1229	6%	2^{-152}	267	
DTRU-1536	1920	2304	-	2^{-196}	405	384 bits
DTRU-2048	2560	3072	-	2^{-204}	561	512 bits
DTRU-Prime	1359	1495	-	2^{-181}	290	256 bits (LPPNF)
SNTRU-Prime-1277	1847	2067	27%	$2^{-\infty}$	270	

and public key sizes, respectively, while Improve Ratio measures size reduction of DTRU at the corresponding security level compared to the scheme in each row. DTRU achieves significantly better compression than NTRU-HPS and NTRU-HRSS, with superior security and lower error rates.

Notably, DAWN prioritizes compression efficiency using sparse ternary distributions, whereas DTRU emphasizes security robustness by avoiding sparse distributions. For fair comparison, we include DTRU-1024$'$ which employs minimal sparsity solely for benchmarking purposes, while our recommended configurations avoid sparse distributions entirely. These schemes exhibit comparable bandwidth, and with the same public key compression techniques, DTRU-Light and DTRU-1024$'$ achieve public key sizes matching DAWN's while maintaining superior security.

Benchmarking on x86 and ARM platforms shows our KEM achieves 3.84-15.69× speedup over NTRU-HRSS and 1.05-1.32× over Kyber in ephemeral key exchange at equivalent security levels (Table 3 in Appendix A).

2 Preliminaries

2.1 Notation

Let $n, q \in \mathbb{N}$ with $q > 0$. Define $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z} \cong \{0, 1, \ldots, q-1\}$. For a monic polynomial Φ of degree n, set $R = \mathbb{Z}[x]/(\Phi)$ and $R_q = \mathbb{Z}_q[x]/(\Phi)$; elements of

these rings are polynomials of degree at most $n-1$ with coefficients in $\mathbb{Z}$ or $\mathbb{Z}_q$, respectively. We denote $f = \sum_{i=0}^{n-1} f_i x^i$ or equivalently $f = (f_0, \ldots, f_{n-1})$.

Throughout, we primarily consider the cyclotomic ring $R_q = \mathbb{Z}_q[x]/(x^n - x^{n/2} + 1)$ with $n = 2^e 3^\ell$. We also consider the LPPNF $R_q = \mathbb{Z}_q[x]/(x^n - x - 1)$ and the power-of-two ring $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ as alternatives.

For $r \in \mathbb{Z}$, let $r' = r \bmod^{\pm} q$ denote the centered representative in $[-\frac{q}{2}, \frac{q}{2})$, and $r' = r \bmod q$ the standard representative in $[0, q)$. For $\omega \in \mathbb{Z}$, define $\|\omega\|_{q,\infty} = |\omega \bmod^{\pm} q|$. For $\omega \in \mathbb{Z}^n$ or R_q, define $\|\omega\|_{q,2} = \sqrt{\sum_{i=0}^{n-1} \|\omega_i\|_{q,\infty}^2}$. The rounding function is denoted by $\lfloor r \rceil = \lfloor r + 1/2 \rfloor$ for $r \in \mathbb{R}$, and is extended coefficient-wise to vectors and polynomials.

We use $x \xleftarrow{\$} D$ to denote uniform sampling from a finite set D, and $x \leftarrow D$ for sampling according to a distribution D. A function $\epsilon : \mathbb{N} \to [0, 1]$ is said to be negligible if for every $c > 0$, there exists λ_0 such that $\epsilon(\lambda) < 1/\lambda^c$ for all $\lambda > \lambda_0$; we denote this by $\mathsf{negl}(\lambda)$.

Centered Binomial Distribution. The centered binomial distribution B_η is defined by sampling $(a_1, \ldots, a_\eta, b_1, \ldots, b_\eta) \in \{0,1\}^{2\eta}$ uniformly at random and outputting $\sum_{i=1}^{\eta}(a_i - b_i)$. Sampling f from B_η indicates that each coefficient is sampled independently from B_η.

Number Theoretic Transform. The Number Theoretic Transform (NTT), the finite-field analogue of FFT, enables efficient polynomial multiplication in $\mathbb{Z}_q$. For $f, g \in R_q$ of degree $n-1$, the product $f \cdot g$ is computed in $O(n \log n)$ time via $\mathrm{INTT}(\mathrm{NTT}(f) \circ \mathrm{NTT}(g))$, where $\circ$ denotes coefficient-wise multiplication. The FFT-trick is a fast algorithm to compute NTT over $\mathbb{Z}_q[x]/(f(x))$ using the Chinese Remainder Theorem (CRT). A radix-N FFT trick corresponds to the isomorphism $\mathbb{Z}_q[x]/(x^{Nm} - \zeta^N) \cong \mathbb{Z}_q[x]/(x^m - \zeta) \times \mathbb{Z}_q[x]/(x^m - \rho\zeta) \times \cdots \times \mathbb{Z}_q[x]/(x^m - \rho^{N-1}\zeta)$, where ρ is a primitive N-th root of unity in $\mathbb{Z}_q$. The forward and inverse transforms employ the Cooley–Tukey (CT) [11] and Gentleman–Sande (GS) [20] butterfly operations, respectively.

2.2 Cryptographic Primitives

Definition 1 (Public-Key Encryption). *A public key encryption scheme (PKE) over message space $\mathcal{M}$ is a tuple of algorithms* $(\mathrm{gen}, \mathrm{enc}, \mathrm{dec})$ *defined as follows:*
(1) $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathrm{gen}(1^\kappa)$*: The key generation algorithm* gen *takes as input the security parameter and outputs a key pair* $(\mathsf{pk}, \mathsf{sk})$*. For brevity, the input to* gen *is often omitted.*
(2) $\mathsf{ct} \leftarrow \mathrm{enc}(\mathsf{pk}, m)$*: The encryption algorithm takes as input the public key* pk*, a message $m \in \mathcal{M}$ and outputs a ciphertext* ct*. If necessary, we make the used randomness explicit by writing* $\mathsf{ct} \leftarrow \mathrm{enc}(\mathsf{pk}, m; \rho)$*.*
(3) $m' \leftarrow \mathrm{dec}(\mathsf{sk}, \mathsf{ct})$*: The decryption algorithm, given the secret key* sk *and ciphertext* ct*, deterministically outputs a message $m' \in \mathcal{M} \cup \{\perp\}$.*

The scheme is δ-correct if $\mathbb{E}\left[\max_{m \in \mathcal{M}} \Pr[\mathrm{dec}(\mathsf{sk}, \mathrm{enc}(\mathsf{pk}, m; \rho)) \neq m]\right] \leq \delta$, where the expectation is over $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathrm{gen}$ and ρ.

Definition 2 (IND-CPA Security). *A PKE scheme is* IND-CPA *secure if for any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, it holds that*

$$\mathrm{Adv}^{\text{IND-CPA}}_{\text{PKE}}(\mathcal{A}) = \left|\Pr[b' = b] - \frac{1}{2}\right| = \mathsf{negl}(\kappa)$$

in the IND-CPA *game in Fig. 1.*

Fig. 1: IND-CPA Game and IND-CCA Game

$\text{IND-CPA}^{\mathcal{A}}_{\text{PKE}}$:	$\text{IND-CCA}^{\mathcal{A}}_{\text{KEM}}$:
1 : $\mathsf{pk}, \mathsf{sk} \leftarrow \mathrm{gen}(1^\kappa)$	1 : $\mathsf{pk}, \mathsf{sk} \leftarrow \mathrm{Gen}(1^\kappa)$
2 : $(m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pk})$	2 : $(\mathsf{ct}^*, K_0^*) \leftarrow \mathrm{Encaps}(\mathsf{pk})$
3 : $b \leftarrow \{0,1\}$	3 : $K_1^* \leftarrow \mathcal{K}$
4 : $\mathsf{ct}_b \leftarrow \mathrm{enc}(\mathsf{pk}, m_b)$	4 : $b \leftarrow \{0,1\}$
5 : $b' \leftarrow \mathcal{A}(\mathsf{ct}_b)$	5 : $b' \leftarrow \mathcal{A}^{O_{\mathrm{Decaps}}(\mathsf{sk}, \cdot \neq \mathsf{ct}^*)}(\mathsf{pk}, \mathsf{ct}^*, K_b^*)$
6 : **return** $b' = b$	6 : **return** $b' = b$

Definition 3 (Key Encapsulation Mechanism). *A key encapsulation mechanism (KEM) over key space $\mathcal{K}$ is a tuple of algorithms* $(\mathrm{Gen}, \mathrm{Encaps}, \mathrm{Decaps})$ *defined as follows:*
(1) $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathrm{Gen}(1^\kappa)$*: The key generation algorithm* Gen *takes as input the security parameter and outputs a key pair* $(\mathsf{pk}, \mathsf{sk})$*. For brevity, the input to* Gen *is often omitted.*
(2) $(\mathsf{ct}, K) \leftarrow \mathrm{Encaps}(\mathsf{pk})$*: The encapsulation algorithm takes as input the public key* pk *and outputs a tuple* (ct, K)*, where $K \in \mathcal{K}$ and $\mathsf{ct} \in \mathcal{C}$.*
(3) $K' \leftarrow \mathrm{Decaps}(\mathsf{sk}, \mathsf{ct})$*: The decapsulation algorithm, given the secret key* sk *and ciphertext* ct*, outputs a key K'. If the KEM supports explicit rejection, the output is $K' \in \mathcal{K}$ or the rejection symbol $\perp$.*

The KEM is δ-correct if $\mathbb{E}[\Pr[\mathrm{Decaps}(\mathsf{sk}, \mathsf{ct}) \neq K, (\mathsf{ct}, K) \leftarrow \mathrm{Encaps}(\mathsf{pk})]] \leq \delta$, where the expectation is over $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathrm{Gen}$ and the randomness of Encaps.

Definition 4 (IND-CCA Security). *A KEM is* IND-CCA *secure if for any PPT adversary $\mathcal{A}$, it holds that*

$$\mathrm{Adv}^{\text{IND-CCA}}_{\text{KEM}}(\mathcal{A}) = \left|\Pr[b' = b] - \frac{1}{2}\right| = \mathsf{negl}(\kappa)$$

in the IND-CCA *game in Fig. 1.*

2.3 Hardness Assumptions

Definition 5 (Decisional NTRU assumption). *Let Ψ_1, Ψ_2 be distributions over a polynomial ring R, and fix an invertible element $p \in R$. Sample $f' \xleftarrow{\$} \Psi_1$ and $g \xleftarrow{\$} \Psi_2$, set $f := 1 + pf'$, and compute $h := gf^{-1} \in R$. The decisional NTRU problem is to distinguish h from a uniformly random element $u \xleftarrow{\$} R$, and the assumption states that the advantage*

$$\mathrm{Adv}^{\mathrm{NTRU}}_{\Psi_1,\Psi_2}(\mathcal{A}) = |\Pr[\mathcal{A}(h) = 1] - \Pr[\mathcal{A}(u) = 1]|$$

for any PPT adversary $\mathcal{A}$ is negligible.

Definition 6 (RLWE assumption). *Let Ψ_1, Ψ_2 be distributions over a polynomial ring R. The (decisional) Ring-Learning with Errors (RLWE) problem over R is to distinguish uniform samples $(u, v) \xleftarrow{\$} R \times R$ from samples $(h, c) \in R \times R$ where $h \xleftarrow{\$} R$ and $c = hr + e$ with $r, e \leftarrow \Psi_1, \Psi_2$. It is hard if the advantage*

$$\mathrm{Adv}^{\mathrm{RLWE}}_{\Psi_1,\Psi_2}(\mathcal{A}) = |\Pr[\mathcal{A}(h, c) = 1] - \Pr[\mathcal{A}(u, v) = 1]|$$

of any PPT adversary $\mathcal{A}$ is negligible.

3 Double E_8 Encoding for NTRU-based KEMs

In this section, we present the double E_8 encoding scheme that underlies the DTRU construction. We begin by reviewing the NEV approach, then describe how we enhance the encoding with lattice coding derived from the E_8 lattice to achieve stronger error correction.

3.1 Motivation

The NEV approach [37] addresses decryption failure issues in NTRU encryption by leveraging repetition encoding. In power-of-two cyclotomic rings $R_q = \mathbb{Z}_q[x]/(x^n + 1)$, NEV employs $f = vf' + 1$ with $v = 1 - x^{n/k}$, and encodes messages using $v^{-1} = \frac{q+1}{2}(1 + x^{n/k} + \cdots + x^{n(k-1)/k})$.

The message $m = m_0 + m_1 x + \cdots + m_{n/k-1} x^{n/k-1}$ is encoded into the top bit(s) of k coefficients, since the coefficients of $v^{-1}m = \frac{q+1}{2}(m + x^{n/k}m + \cdots + x^{n(k-1)/k}m)$ are distributed as:

$$\frac{q+1}{2}(\overbrace{m_0, m_1, \cdots, m_{\frac{n}{k}-1}}^{m}, \overbrace{m_0, m_1, \cdots, m_{\frac{n}{k}-1}}^{x^{n/k}m}, \cdots, \overbrace{m_0, m_1, \cdots, m_{\frac{n}{k}-1}}^{x^{n(k-1)/k}m}).$$

This repetition encoding distributes each message bit across k coefficients, significantly enhancing error correction capabilities.

While NEV's repetition encoding provides substantial improvements, it relies solely on this basic coding property. The approach can be further strengthened by

incorporating additional coding techniques. Specifically, the message $m \in \{0,1\}^l$ can be first mapped via lattice coding to $w \in \{0,1\}^{n/k}$, then encoded as $v^{-1}w$. By deferring the multiplication by the coefficient $\frac{q+1}{2}$ of v^{-1} to the final step, the preceding encoding process effectively forms a binary encoding.

Notably, this encoding mechanism extends beyond power-of-two cyclotomic rings, even in rings lacking specific v properties, setting $f = pf' + 1$ with $p = 2$ preserves the core coding benefits.

This insight motivates our construction, which aims to enhance error correction by integrating more sophisticated coding schemes while extending applicability to a broader class of rings.

3.2 Double E_8 Encoding

Our construction, DTRU, aims to encode $n/4$ bits into n coefficients. This encoding requirement necessitates a binary code that maps 1 bit to 2 bits on average. When combined with NEV's repetition code, this yields a 1-to-4 mapping.

Building upon this foundation, we further enhance error correction by increasing the encoding dimension. Specifically, we employ a four-dimensional sublattice of the E_8 lattice to construct an extended Hamming-style code that maps 4 bits to 8 bits, achieving a minimum Hamming distance of 4, an optimal property among binary [8,4] codes. The combination of this extended Hamming-style code with NEV's repetition encoding yields a double E_8 encoding that effectively maps 4 bits to 16 coordinates while maintaining modest decoding complexity.

Message Encoding The core component of our double E_8 encoding is the E_8 code, which is constructed from a four-dimensional sublattice of the E_8 lattice. To encode a message $m \in \{0,1\}^4$, we compute the codeword $u = mH \bmod 2$, where H is the generator matrix defined as follows:

$$H = \begin{bmatrix} 1\,1\,1\,1\,0\,0\,0\,0 \\ 0\,0\,1\,1\,1\,1\,0\,0 \\ 0\,0\,0\,0\,1\,1\,1\,1 \\ 0\,1\,0\,1\,0\,1\,0\,1 \end{bmatrix}.$$

This E_8 code achieves a minimum Hamming distance of $d = 4$, which is optimal among binary [8,4] codes.

The double E_8 encoding repeats each E_8 codeword. Specifically, a message $m \in \{0,1\}^4$ is first encoded as $u = mH \bmod 2$, then repeated to form $(u,u) \in \{0,1\}^{16}$, effectively mapping 4 bits to 16 coordinates. This combination yields a code of length 16 and dimension 4 with minimum Hamming distance $d = 8$. The 16 logical coordinates need not occupy 16 consecutive polynomial coefficients: their physical placement is chosen according to the covariance structure induced by the defining polynomial of the ring.

In our encryption algorithm, the double E_8 code is scaled by $\frac{q+1}{2}$ to operate over $\mathbb{Z}_q$. We can establish that the minimum Euclidean distance between any two distinct scaled codewords is at least $\sqrt{2}(q-1)$. This distance property arises

because for any two distinct codewords $\frac{q+1}{2}(u_1, u_1)$ and $\frac{q+1}{2}(u_2, u_2)$, the difference $(u_1, u_1) - (u_2, u_2)$ in $\mathbb{Z}_2^{16}$ is a nonzero codeword of the double E_8 code with Hamming weight at least 8. Considering the Euclidean norm in $\mathbb{Z}_q$, we obtain

$$\left\| \frac{q+1}{2}\big((u_1, u_1) - (u_2, u_2)\big) \right\|_{q,2} \geq \frac{q-1}{2}\sqrt{8} = \sqrt{2}(q-1).$$

We next specify this ring-dependent placement. Let $B = \lceil \ell/4 \rceil$ and pad the message with zeros to length $4B$. For block $i \in \{0, \ldots, B-1\}$ and logical coordinate $j \in \{0, \ldots, 15\}$, define

$$\pi_R(i,j) = \begin{cases} 8i + j, & R_q = \mathbb{Z}_q[x]/(x^n + 1),\ 0 \leq j < 8, \\ n/2 + 8i + (j-8), & R_q = \mathbb{Z}_q[x]/(x^n + 1),\ 8 \leq j < 16, \\ 16i + j, & R_q = \mathbb{Z}_q[x]/(x^n - x^{n/2} + 1), \\ 32\lfloor i/2 \rfloor + 2j + (i \bmod 2), & R_q = \mathbb{Z}_q[x]/(x^n - x - 1). \end{cases} \tag{1}$$

The supported message lengths satisfy $\max_{i<B,\, j<16} \pi_R(i,j) < n$. That this placement avoids potentially correlated coefficient pairs within each decoding block can be verified using the covariance characterization in Proposition 3 of [38]. In the power-of-two cyclotomic ring, distinct coefficients are uncorrelated; the first eight coordinates of every block form a polynomial w' of degree less than $n/2$, and multiplication by $1 + x^{n/2}$ produces the repeated half, so that $w = (1 + x^{n/2})w'$. In the tricyclotomic ring, potentially correlated coefficient pairs are separated by $n/2$; therefore, all 16 coordinates of a block are placed consecutively. In the LPPNF, potential correlation occurs between neighboring coefficients, so two blocks are interleaved: one occupies the even positions and the other the odd positions of each 32-coefficient segment. Thus, no two neighboring polynomial coefficients belong to the same decoding block. This odd-even interleaving is repeated throughout the polynomial.

Algorithm 1 gives the resulting unified encoder. Unused polynomial coefficients are set to zero.

Message Decoding Our decoding algorithm, formalized in Algorithm 2, is based on the maximum-likelihood decoding principle. Given a received polynomial $t \in R_q$, it reconstructs each 16-dimensional decoding block as $v = (t_{\pi_R(i,0)}, \ldots, t_{\pi_R(i,15)})$. Thus, decoding uses exactly the inverse of the ring-dependent coefficient placement used by the encoder. For each block, the objective is to find a codeword $\frac{q+1}{2}(u, u)$ that minimizes the squared Euclidean distance $\|v - \frac{q+1}{2}(u, u)\|_{q,2}^2$.

To identify the most likely transmitted message, we evaluate the distance metric for all possible codewords. For each candidate codeword, the total distance is obtained by summing the relevant entries from the distance matrix according to the codeword's structure. The message $m \in \{0,1\}^4$ corresponding to the codeword that minimizes this total distance is selected as the decoded message.

Algorithm 1 Encode

Require: the message $m \in \{0,1\}^{\ell}$ with $\max_{i<B,\, j<16} \pi_R(i,j) < n$
Ensure: the encoded polynomial $w \in R_q$
$B \leftarrow \lceil \ell/4 \rceil$; pad m with zeros to length $4B$
$w_0, \ldots, w_{n-1} \leftarrow 0$
for $i = 0$ to $B-1$ **do**
  $(u_0, \ldots, u_7) \leftarrow (m_{4i}, m_{4i+1}, m_{4i+2}, m_{4i+3})H \bmod 2$
  for $j = 0$ to 15 **do**
    $w_{\pi_R(i,j)} \leftarrow u_{j \bmod 8}$
  end for
end for
return $w_0 + w_1 x + \cdots + w_{n-1}x^{n-1}$ in R_q

Algorithm 2 Decode

Require: $t = t_0 + t_1 x + \cdots + t_{n-1}x^{n-1} \in R_q$
Ensure: $m \in \{0,1\}^{\ell}$
$B \leftarrow \lceil \ell/4 \rceil$
for $i = 0$ to $B-1$ **do**
  for $j = 0$ to 15 **do**
    $v_j \leftarrow t_{\pi_R(i,j)}$
  end for
  for $j = 0$ to 7 **do**
    dis_matrix$[j][0] \leftarrow \|v_j\|_{q,2}^2 + \|v_{j+8}\|_{q,2}^2$
    dis_matrix$[j][1] \leftarrow \|v_j - \frac{q+1}{2}\|_{q,2}^2 + \|v_{j+8} - \frac{q+1}{2}\|_{q,2}^2$
  end for
  for $j = 0$ to 15 **do**
    $(u_0, \ldots, u_7) \leftarrow (j_0, j_1, j_2, j_3) \cdot H$ ▷ $j = (j_0 j_1 j_2 j_3)_2$
    cost$[j] \leftarrow \sum_{k=0}^{7}$ dis_matrix$[k][u_k]$
  end for
  $k \leftarrow \arg\min$ cost
  $(m_{4i}, m_{4i+1}, m_{4i+2}, m_{4i+3}) \leftarrow (k_0, k_1, k_2, k_3)$ ▷ $k = (k_0 k_1 k_2 k_3)_2$
end for
return m ▷ Remove the zero padding beyond position $\ell - 1$

The correctness of this decoding procedure is guaranteed by the minimum distance property of the scaled double E_8 code. With a minimum Euclidean distance of $\sqrt{2}(q-1)$, if the noise magnitude satisfies

$$\left\| v - \frac{q+1}{2}(u,u) \right\|_{q,2} < \frac{q-1}{\sqrt{2}},$$

the closest codeword to the received vector must be the transmitted codeword, ensuring correct decoding.

4 NTRU Encryption Scheme with Double E_8 Encoding

In this section, we present our NTRU encryption scheme with double E_8 encoding, referred to as DTRU.PKE, as well as its corresponding key encapsulation mechanism, DTRU.KEM. Furthermore, we analyze the error rate of the scheme and establish its provable security under standard cryptographic assumptions.

4.1 Algorithms of DTRU.PKE with Double E_8 encoding

The algorithms for key generation, encryption, and decryption in our DTRU public key encryption scheme are formally presented in Fig. 2.

Fig. 2: Algorithms of DTRU.PKE

DTRU.PKE.KeyGen:

1 : $f', g \leftarrow (\Psi_f, \Psi_g)$
2 : $f \leftarrow pf' + 1$
3 : **if** f is not invertible in R_q
4 : go to step 1 and resample f'
5 : **endif**
6 : $h \leftarrow gf^{-1}$ in R_q
7 : **return** $(\mathsf{pk}, \mathsf{sk}) = (h, f)$

DTRU.PKE.Enc($\mathsf{pk} = h, m \in \{0,1\}^\ell$):

1 : $r, e \leftarrow (\Psi_r, \Psi_e)$
2 : $w \leftarrow \text{Encode}(m)$
3 : $c \leftarrow \lfloor \frac{q_2}{q}(hr + e + \frac{q+1}{2}w) \rceil \bmod q_2$
4 : **return** c

DTRU.PKE.Dec($\mathsf{sk} = f, c$):

1 : $c' \leftarrow \lfloor \frac{q}{q_2} \cdot c \rceil \bmod q$
2 : $m' \leftarrow \text{Decode}(c'f)$
3 : **return** m'

The DTRU.PKE scheme follows a standard NTRU-based encryption framework with several key innovations. The algorithm flow consists of three main components:

Key Generation: The scheme generates a key pair $(\mathsf{pk}, \mathsf{sk}) = (h, f)$, where $f = pf' + 1$ is constructed to ensure invertibility in the ring R_q. The parameter p is chosen based on the ring structure: $p = 2$ for tricyclotomic rings and LPPNF, or $p = 1 - x^{n/2}$ for power-of-two cyclotomic rings. This parameter selection is specifically designed to minimize the error rate, as analyzed in the subsequent error rate analysis.

Encryption: The encryption process samples noise polynomials r and e, encodes the message m using double E_8 encoding, and computes the ciphertext. A key feature is the compression operation $\lfloor \frac{q_2}{q}(\cdot) \rceil \bmod q_2$, which reduces ciphertext size by compressing from modulus q to q_2.

Decryption: Decryption reverses the compression to recover the original modulus, then uses the secret key f to recover the encoded message, which is decoded back to the plaintext.

4.2 Analysis of Error Rate

Theorem 1. *Let $\Psi_f, \Psi_g, \Psi_r, \Psi_e$ be distributions over the ring R_q, with f', g, r, e sampled accordingly. Let $p = 2$ (or $p = 1 - x^{n/2}$), $f = pf' + 1$, and $w = \mathrm{Encode}(m)$ encode a message $m \in \mathcal{M}$, with ϵ denoting the rounding error from ciphertext compression and decompression. For the power-of-two cyclotomic ring, write $w = (1 + x^{n/2})w'$ with $\deg(w') < n/2$. Let $Err = gr + f(e + \epsilon) + f'w$ in the tricyclotomic ring and LPPNF, and let $Err = gr + f(e + \epsilon) + f'w'$ in the power-of-two cyclotomic ring. Define the noise vector of decoding block i by*

$$\mathbf{Err}^{(i)} = (Err_{\pi_R(i,0)}, \ldots, Err_{\pi_R(i,15)}).$$

If

$$1 - \delta = \Pr\left[\|\mathbf{Err}^{(i)}\|_{q,2}^2 < (q-1)^2/2 \text{ for every } 0 \le i < B\right],$$

then the decryption failure rate is bounded above by δ.

Proof. The decryption algorithm yields $m' = \mathrm{Decode}\,(c'f \bmod q)$. In the power-of-two cyclotomic ring, $w = \mathrm{Encode}(m) = (1 + x^{n/2})w'$ for a polynomial w' of degree less than $n/2$; in the other two rings, w is formed directly according to the placement map (1).

Let $\epsilon = \left\lfloor \frac{q}{q_2} \left\lfloor \frac{q_2}{q}(hr + e + \frac{q+1}{2}w) \right\rceil \right\rceil - (hr + e + \frac{q+1}{2}w)$ be the rounding error. Then

$$c' = \left\lfloor \frac{q}{q_2} c \right\rceil = \left\lfloor \frac{q}{q_2} \left\lfloor \frac{q_2}{q}(hr + e + \frac{q+1}{2}w) \right\rceil \right\rceil = hr + e + \frac{q+1}{2}w + \epsilon.$$

Since

$$\begin{aligned} \frac{q+1}{2}w \cdot f = \frac{q+1}{2}w \cdot (pf' + 1) &= \frac{q+1}{2}(1 + x^{n/2})w' \cdot pf' + \frac{q+1}{2}w \\ &= \begin{cases} f'w' + \dfrac{q+1}{2}w & p = 1 - x^{n/2} \text{ in } R_q = \mathbb{Z}_q[x]/(x^n + 1), \\ f'w + \dfrac{q+1}{2}w & p = 2 \text{ in other rings.} \end{cases} \end{aligned}$$

We have

$$c'f = \begin{cases} \underbrace{gr + f(e + \epsilon) + f'w'}_{Err} + \frac{q+1}{2}w & p = 1 - x^{n/2} \text{ in } R_q = \mathbb{Z}_q[x]/(x^n + 1), \\ \underbrace{gr + f(e + \epsilon) + f'w}_{Err} + \frac{q+1}{2}w & p = 2 \text{ in other rings.} \end{cases}$$

Under the placement map (1), the noise affecting the i-th scaled codeword is exactly $\mathbf{Err}^{(i)}$. By the minimum-distance property of the scaled code, if

$$\|\mathbf{Err}^{(i)}\|_{q,2}^2 < \left(\frac{q-1}{\sqrt{2}}\right)^2,$$

then the i-th codeword decodes correctly.

Hence, decryption succeeds when $\|\mathbf{Err}^{(i)}\|_{q,2}^2 < \frac{(q-1)^2}{2}$ for every $0 \le i < B$, and the failure probability is bounded above by δ.

To estimate the decryption failure probability δ, we treat the 16 coefficients of each decoding block as asymptotically independent. We justify this approximation from the following three perspectives.

Theoretical justification. The approximation follows from joint Gaussian convergence together with the covariance structure of the selected coefficients. For the power-of-two cyclotomic ring, Theorem 1 and Lemma 1 of [29] establish the relevant asymptotic multivariate Gaussian model. For the tricyclotomic ring and LPPNF, an analogous result can be proved by applying the same multivariate central-limit argument to their respective polynomial-reduction patterns and coefficient selections: after normalization, the vector formed by the coefficients in any fixed 16-coordinate block converges jointly to a centered multivariate Gaussian distribution as n tends to infinity.

It remains to examine the limiting covariance matrices. Proposition 3 of [38] characterizes the relevant covariance structures: distinct coefficients are uncorrelated in the power-of-two cyclotomic ring; in the tricyclotomic ring, nonzero covariance can occur only for coefficients whose indices differ by $n/2$; and in the LPPNF, it can occur only for neighboring coefficients. The placement map (1) excludes all such potentially correlated pairs from any individual 16-coordinate block. Consequently, the covariance matrix of each block is diagonal. Since a jointly Gaussian vector with diagonal covariance has independent coordinates, the 16 coefficients in each decoding block are asymptotically independent. Our DFR tests use these same coefficient orderings.

Precedent in established schemes. Kyber [34] and Scloud$^+$ [36] adopt analogous independence assumptions when estimating their failure rates, treating rounding errors as i.i.d. and independent of other terms (see also Section 2.3 in [29]).

We stress that fully eliminating the independence assumption for high-dimensional lattice codes remains an open problem [33], and contemporary schemes such as NEV and DAWN rely on the same assumption. Under these circumstances, adopting the independence assumption is both a pragmatic necessity and a well-motivated choice.

Under the independent Gaussian approximation, let $\widetilde{Err}_{\pi_R(i,j)} \in \mathbb{Z}$ denote the coefficient before reduction modulo q, after reduction by the defining polynomial of the ring, and define

$$v_{i,j} = \mathrm{Var}\left(\widetilde{Err}_{\pi_R(i,j)}\right), \qquad \widetilde{Q}_i = \sum_{j=0}^{15} \widetilde{Err}^2_{\pi_R(i,j)}, \qquad Q_i = \|\mathbf{Err}^{(i)}\|^2_{q,2}.$$

Since $|a \bmod^{\pm} q| \le |a|$ for every $a \in \mathbb{Z}$, we have $Q_i \le \widetilde{Q}_i$. Thus, the modular norm used by the decoder is fully accounted for, while the unreduced norm provides a conservative upper bound on the failure event. Under the Gaussian approximation,

$$\widetilde{Q}_i \overset{d}{\approx} \sum_{j=0}^{15} v_{i,j} Z^2_{i,j}, \qquad Z_{i,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

For the tricyclotomic ring and LPPNF, the variances $v_{i,j}$ within one block need not be equal. Our DFR evaluation therefore matches the first two moments of this generalized chi-squared variable with a scaled chi-squared variable $\lambda_i \chi^2(\nu_i)$. Writing

$$S_{1,i} = \sum_{j=0}^{15} v_{i,j}, \qquad S_{2,i} = \sum_{j=0}^{15} v_{i,j}^2,$$

the moment-matched parameters are

$$\nu_i = \frac{S_{1,i}^2}{S_{2,i}}, \qquad \lambda_i = \frac{S_{2,i}}{S_{1,i}}, \qquad \widetilde{Q}_i \overset{d}{\approx} \lambda_i \chi^2(\nu_i).$$

Equivalently, this is a Gamma distribution with shape $\nu_i/2$ and scale $2\lambda_i$. Applying the union bound gives

$$\begin{aligned} \delta &\leq \sum_{i=0}^{B-1} \Pr\left[Q_i \geq \frac{(q-1)^2}{2}\right] \\ &\leq \sum_{i=0}^{B-1} \Pr\left[\widetilde{Q}_i \geq \frac{(q-1)^2}{2}\right] \\ &\approx \sum_{i=0}^{B-1} \Pr\left[\chi^2(\nu_i) \geq \frac{(q-1)^2}{2\lambda_i}\right]. \end{aligned}$$

In the power-of-two cyclotomic ring, all 16 variances in a block are equal to a common value σ_{Err}^2. Hence $\nu_i = 16$ and $\lambda_i = \sigma_{Err}^2$, recovering the usual approximation $\widetilde{Q}_i \approx \sigma_{Err}^2 \chi^2(16)$.

Our decryption failure rate (DFR) analysis, detailed in Table 2, demonstrates that the selected parameters achieve failure probabilities in the range of 2^{-134} to 2^{-204} across different security levels. This conservative parameter design ensures that even under worst-case conditions (where an adversary crafts the most challenging message to induce high DFR), the failure rates increase by no more than a factor of approximately 2^{20} compared to average-case scenarios, thereby effectively mitigating the risk of chosen ciphertext attacks that exploit failure rate disparities [12,22].

4.3 Security of DTRU.PKE

Theorem 2. *Let $\Psi_f, \Psi_g, \Psi_r, \Psi_e$ be the distributions over the ring R_q. Then, under the Decisional NTRU assumption and RLWE assumption, our DTRU.PKE scheme is IND-CPA secure in the standard model. That is, for any probabilistic polynomial-time adversary $\mathcal{A}$, there exist adversaries $\mathcal{B}$ and $\mathcal{C}$ such that* $\mathrm{Adv}^{\text{IND-CPA}}_{\text{DTRU.PKE}}(\mathcal{A}) \leq \mathrm{Adv}^{\text{NTRU}}_{\Psi_f,\Psi_g}(\mathcal{B}) + \mathrm{Adv}^{\text{RLWE}}_{\Psi_r,\Psi_e}(\mathcal{C})$.

The proof of Theorem 2 is given in Appendix B.

4.4 Algorithms of DTRU.KEM

We construct DTRU.KEM by using the $\mathrm{FO}^{\not\perp}_{ID(\mathsf{pk}),m}$ transformation [15].

Let $\mathcal{M} = \{0,1\}^{\ell}$ be the message space of DTRU.PKE. We define a hash function $H : \{0,1\}^* \to \{0,1\}^{\kappa} \times \{0,1\}^{\mathrm{coins}}$, where coins denotes the randomness length for DTRU.PKE.Enc, and a fixed-output-length function $\mathrm{ID} : \mathcal{PK} \to \{0,1\}^{\gamma}$ (with $\gamma \geq 256$ for security) that extracts a fixed-length prefix from public keys. The DTRU.KEM algorithms are presented in Fig. 3.

Fig. 3: Algorithms of DTRU.KEM

DTRU.KEM.KeyGen:

$(\mathsf{pk}, \mathsf{sk}') \leftarrow \mathrm{DTRU.PKE.KeyGen}(1^{\kappa})$
$s \leftarrow \{0,1\}^{\kappa}$
return $(\mathsf{pk}, \mathsf{sk} = (\mathsf{sk}', \mathsf{pk}, s))$

DTRU.KEM.Encaps:

$m \leftarrow \mathcal{M}$
$(K, \rho) \leftarrow H(\mathrm{ID}(\mathsf{pk}), m)$
$c \leftarrow \mathrm{DTRU.PKE.Enc}(\mathsf{pk}, m; \rho)$
return (c, K)

DTRU.KEM.Decaps:

$m' \leftarrow \mathrm{DTRU.PKE.Dec}(\mathsf{sk}', c)$
$(K', \rho') \leftarrow H(\mathrm{ID}(\mathsf{pk}), m')$
$\tilde{K} \leftarrow H_1(\mathrm{ID}(\mathsf{pk}), s, c)$
if $\mathrm{DTRU.PKE.Enc}(\mathsf{pk}, m'; \rho') \neq c$
  return $\tilde{K}$
else
  return K'
endif

Since DTRU.PKE is IND-CPA secure, by the security proof of $\mathrm{FO}^{\not\perp}_{ID(\mathsf{pk}),m}$ transformation in [15], we have the following theorem.

Theorem 3. *Let γ be the output length of ID and ℓ be the min entropy of $ID(\mathsf{pk})$, i.e. $\ell = H_{\infty}(ID(\mathsf{pk}))$, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathrm{DTRU.PKE.KeyGen}$. For any (quantum) adversary $\mathcal{A}$ that makes at most q_D decapsulation queries and q_H (quantum) random oracle queries against the IND-CCA security of DTRU.KEM, there exists a (quantum) adversary $\mathcal{B}$ with roughly the same running time as $\mathcal{A}$ such that:*

- *In the random oracle model (ROM),* $\mathrm{Adv}^{\text{IND-CCA}}_{\mathrm{DTRU.KEM}}(\mathcal{A}) \leq$

$$2\left(\mathrm{Adv}^{\text{IND-CPA}}_{\mathrm{DTRU.PKE}}(\mathcal{B}) + \frac{q_H + 1}{|\mathcal{M}|}\right) + \frac{q_H}{2^{\kappa}} + (q_H + q_D)\delta + \frac{1}{2^{\ell}}$$

- *In the quantum random oracle model (QROM),* $\mathrm{Adv}^{\text{IND-CCA}}_{\mathrm{DTRU.KEM}}(\mathcal{A}) \leq$

$$2\sqrt{q\mathrm{Adv}^{\text{IND-CPA}}_{\mathrm{DTRU.PKE}}(\mathcal{B})} + \frac{4q}{\sqrt{|\mathcal{M}|}} + \frac{1}{2^{\ell}} + \frac{4(q_H + 1)}{\sqrt{2^{\kappa}}} + 16q^2\delta + \frac{1}{|\mathcal{M}|},$$

where $q = q_H + q_D + 1$ and δ denotes the decryption failure probability of DTRU.PKE.

5 Concrete Security and Parameter Selection

This section analyzes the security of the DTRU scheme, which relies on the computational hardness of RLWE and NTRU problems. We evaluate their resilience against state-of-the-art cryptanalytic techniques.

5.1 Concrete Attacks on RLWE

To evaluate the security strength of the RLWE problem, we adopt the Core-SVP framework and analyze various attack methodologies.

Since standard lattice reduction algorithms like BKZ cannot directly exploit the algebraic structure of RLWE, except for certain weak instances identified in [8,16], we first transform the RLWE instance into an equivalent LWE instance for analysis.

Given an RLWE instance $(a, t = as + e) \in R_q \times R_q$, we transform it to an LWE instance as follows: Let $\mathbf{a} \in \mathbb{Z}_q^n$ be the coefficient vector of polynomial a, and define the matrix $A \in \mathbb{Z}_q^{n \times n}$ as the matrix representation of multiplication by a in the ring R_q. Similarly, let $\mathbf{s}, \mathbf{e}, \mathbf{t} \in \mathbb{Z}_q^n$ be the coefficient vectors of s, e, t respectively. Then the RLWE instance becomes the LWE instance:

$$(A, \mathbf{t} = A\mathbf{s} + \mathbf{e}) \in \mathbb{Z}_q^{n \times n} \times \mathbb{Z}_q^n.$$

Core-SVP Methodology. BKZ-class attacks [9] represent the most effective methods against RLWE. These attacks transform the LWE problem into a short vector problem (SVP) on a lattice using embedding techniques (Kannan embedding [24] and Bai-Galbraith embedding [5]), then solve the SVP using the BKZ lattice basis reduction algorithm.

The BKZ algorithm iteratively examines b-dimensional projected sublattices, solving the SVP on these sublattices using enumeration or sieving. The reduced basis vectors follow the geometric series assumption, enabling estimation of the block size b required to solve the SVP, which provides the security estimate. For our security assessment, we utilize the modern cost framework presented in [32].
Primal Attack. The primal attack reduces LWE to solving the unique shortest vector problem (u-SVP). Given $(A, \mathbf{t} = A\mathbf{s} + \mathbf{e})$, we construct an $(n + m + 1)$-dimensional lattice:

$$\mathcal{L} = \left\{ \mathbf{x} \in \mathbb{Z}^{n+m+1} \mid (A_{[1:m]} \mid I_m \mid -\mathbf{t}_{[1:m]})\mathbf{x} \equiv \mathbf{0} \pmod{q} \right\}$$

where $m \leq n$. The lattice basis matrix is:

$$B = \begin{pmatrix} I_n & 0 & 0 \\ -A_{[1:m]} & qI_m & \mathbf{t}_{[1:m]} \\ 0 & 0 & 1 \end{pmatrix}$$

The unique shortest vector $\mathbf{v} = (\mathbf{s}^T \mid \mathbf{e}^T \mid 1)^T$ in $\mathcal{L}$ solves the LWE problem, with $\|\mathbf{v}\| \approx \sqrt{\alpha_1^2 n + \alpha_2^2 m}$.

Dual Attack. The dual attack finds short vectors in the dual lattice. Given $(A, \mathbf{t})$, we construct the dual lattice:

$$\mathcal{L}^* = \left\{ (\mathbf{x}, \mathbf{y}) \in \mathbb{Z}^m \times \mathbb{Z}^n \mid A^T_{[1:m]}\mathbf{x} = \mathbf{y} \pmod q \right\}$$

A short vector $(\mathbf{x}, \mathbf{y}) \in \mathcal{L}^*$ allows recovery of $\mathbf{s}_{[m+1:n]}$ through:

$$\langle \mathbf{x}, \mathbf{t} \rangle = \langle \mathbf{x}, \mathbf{e} \rangle + \langle \mathbf{y}, \mathbf{s}_{[1:m]} \rangle + \langle A^T_{[m+1:n]}\mathbf{x}, \mathbf{s}_{[m+1:n]} \rangle.$$

BKW attack. The Blum-Kalai-Wasserman (BKW) algorithm employs combinatorial techniques rather than lattice reduction. Recent composite approaches [21,27] have enhanced BKW's efficiency, making it the most effective attack for specific parameter regimes.

Algebraic attack. Algebraic attacks, introduced in [4], construct equations for the variables $s_{[1:m]}$ from the small set $e_{[1:m]}$, yielding n high-degree equations in n variables. These are typically solved using Gröbner basis computations. In practice, however, algebraic approaches often provide limited advantage.

5.2 Concrete Attacks on NTRU

Similar to the RLWE instance, the NTRU instance $(h = f^{-1}g \mod q) \in R_q$ can be viewed as a RLWE instance $(h, 0 = hf - g) \in R_q \times R_q$ with a zero target vector, which can be transformed into the corresponding LWE instances $(H', 0 = H'f - g) \in \mathbb{Z}_q^{n \times n} \times \mathbb{Z}_q^n$ for analysis.

While most attack methods on RLWE can be applied to NTRU, the dual attack [1] is not applicable. The dual attack relies on a non-zero target vector t' to recover the secret through the computation $\langle x, t' \rangle = \langle x, e \rangle + \langle y, s \rangle$. In the NTRU-to-LWE transformation, the target vector becomes zero, rendering this mechanism ineffective.

5.3 Concrete Security of DTRU and Parameter Selection

We employ the aforementioned attack methodologies to evaluate the security strength of the DTRU scheme using the lattice estimator[2] from [2].

The results are summarized in Table 2. The DTRU scheme employs tricyclotomic rings, DTRU-Light uses power-of-two cyclotomic rings, and DTRU-Prime is based on LPPNF. Parameters n, q, and q_2 represent the dimension, modulus, and compressed modulus, respectively. The distribution parameters (k_g, k_f) and (k_r, k_e) correspond to key generation (f, g) and encryption (r, e), respectively. The (C, Q) values for LWE and NTRU columns indicate security levels against classical and quantum attacks, while δ denotes the decryption failure rate under the specified parameters.

[2] Available at https://github.com/malb/lattice-estimator. All security estimates are computed with the estimator's default settings: (1) BKZ cost model MATZOV [32], which accounts for amortized list-decoding sieving costs under the classical BDGL16 algorithm; (2) both primal and dual attacks are evaluated with internal parameter optimization to minimize the total attack cost; (3) all other parameters remain at their default values as specified in [2].

Table 2: Parameters and Security Levels of DTRU Schemes

Scheme	n	q	q_2	(k_g, k_f)	(k_r, k_e)	LWE (C,Q)	NTRU (C,Q)	δ
DTRU	512	3457	2^{10}	(B_8, B_4)	(B_6, B_3)	$(150, 144)$	$(151, 144)$	$2^{-134.26}$
	768	3457	2^{10}	(B_4, B_3)	(B_4, B_2)	$(213, 203)$	$(213, 200)$	$2^{-185.56}$
	1024	3457	2^{10}	(B_5, B_2)	(B_3, B_2)	$(280, 261)$	$(284, 263)$	$2^{-190.48}$
	1536	3457	2^{10}	(B_3, B_2)	(B_2, B_1)	$(405, 376)$	$(422, 388)$	$2^{-195.50}$
	2048	3457	2^{10}	(B_5, B_1)	(B_2, B_1)	$(574, 524)$	$(561, 512)$	$2^{-204.10}$
DTRU-Light	512	769	2^8	(B_2, B_1)	(B_1, B_2)	$(152, 147)$	$(151, 144)$	$2^{-143.92}$
DTRU-Prime	1087	2017	2^{10}	(B_2, B_1)	(B_2, B_1)	$(293, 274)$	$(290, 269)$	$2^{-180.72}$

References

1. Albrecht, M.R., Curtis, B.R., Deo, A., Davidson, A., Player, R., Postlethwaite, E.W., Virdia, F., Wunderer, T.: Estimate all the {LWE, NTRU} schemes! In: SCN. Lecture Notes in Computer Science, vol. 11035, pp. 351–367. Springer (2018)
2. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. J. Math. Cryptol. **9**(3), 169–203 (2015)
3. Alkim, E., Cheng, D.Y., Chung, C.M., Evkan, H., Huang, L.W., Hwang, V., Li, C.T., Niederhagen, R., Shih, C., Wälde, J., Yang, B.: Polynomial multiplication in NTRU prime comparison of optimization strategies on cortex-m4. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2021**(1), 217–238 (2021)
4. Arora, S., Ge, R.: New algorithms for learning in presence of errors. In: ICALP (1). Lecture Notes in Computer Science, vol. 6755, pp. 403–415. Springer (2011)
5. Bai, S., Galbraith, S.D.: Lattice decoding attacks on binary LWE. In: ACISP. Lecture Notes in Computer Science, vol. 8544, pp. 322–337. Springer (2014)
6. Bai, S., Jangir, H., Lin, H., Ngo, T., Wen, W., Zheng, J.: Compact encryption based on module-ntru problems. In: PQCrypto (1). Lecture Notes in Computer Science, vol. 14771, pp. 371–405. Springer (2024)
7. Bernstein, D.J., Chuengsatiansup, C., Lange, T., van Vredendaal, C.: NTRU prime: Reducing attack surface at low cost. In: SAC. Lecture Notes in Computer Science, vol. 10719, pp. 235–260. Springer (2017)
8. Castryck, W., Iliashenko, I., Vercauteren, F.: Provably weak instances of ring-lwe revisited. In: EUROCRYPT (1). pp. 147–167. Lecture Notes in Computer Science, Springer (2016)
9. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: ASIACRYPT. Lecture Notes in Computer Science, vol. 7073, pp. 1–20. Springer (2011)
10. Chung, C.M., Hwang, V., Kannwischer, M.J., Seiler, G., Shih, C., Yang, B.: NTT multiplication for ntt-unfriendly rings new speed records for saber and NTRU on cortex-m4 and AVX2. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2021**(2), 159–188 (2021)
11. Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex fourier series. Mathematics of Computation **19**, 297–301 (1965), https://api.semanticscholar.org/CorpusID:121744946
12. D'Anvers, J., Guo, Q., Johansson, T., Nilsson, A., Vercauteren, F., Verbauwhede, I.: Decryption failure attacks on IND-CCA secure lattice-based schemes. In: Public Key Cryptography (2). pp. 565–598. Lecture Notes in Computer Science, Springer (2019)

13. Ducas, L., Lyubashevsky, V., Prest, T.: Efficient identity-based encryption over NTRU lattices. In: ASIACRYPT (2). Lecture Notes in Computer Science, vol. 8874, pp. 22–41. Springer (2014)
14. Ducas, L., van Woerden, W.P.J.: NTRU fatigue: How stretched is overstretched? In: ASIACRYPT (4). Lecture Notes in Computer Science, vol. 13093, pp. 3–32. Springer (2021)
15. Duman, J., Hövelmanns, K., Kiltz, E., Lyubashevsky, V., Seiler, G.: Faster lattice-based kems via a generic fujisaki-okamoto transform using prefix hashing. In: ACM CCS 2021. pp. 2722–2737 (2021)
16. Elias, Y., Lauter, K.E., Ozman, E., Stange, K.E.: Provably weak instances of ring-lwe. In: CRYPTO (1). pp. 63–92. Lecture Notes in Computer Science, Springer (2015)
17. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z., et al.: Falcon: Fast-fourier lattice-based compact signatures over ntru. Submission to the NIST's post-quantum cryptography standardization process **36**(5), 1–75 (2018)
18. Fouque, P., Kirchner, P., Pornin, T., Yu, Y.: BAT: small and fast KEM over NTRU lattices. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2022**(2), 240–265 (2022)
19. Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 7881, pp. 1–17. Springer (2013)
20. Gentleman, W.M., Sande, G.: Fast fourier transforms: for fun and profit. In: AFIPS Fall Joint Computing Conference. AFIPS Conference Proceedings, vol. 29, pp. 563–578. AFIPS / ACM / Spartan Books, Washington D.C. (1966)
21. Guo, Q., Johansson, T., Stankovski, P.: Coded-bkw: Solving LWE using lattice codes. In: CRYPTO (1). Lecture Notes in Computer Science, vol. 9215, pp. 23–42. Springer (2015)
22. Guo, Q., Johansson, T., Yang, J.: A novel CCA attack using decryption errors against LAC. In: ASIACRYPT (1). pp. 82–111. Lecture Notes in Computer Science, Springer (2019)
23. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: ANTS. Lecture Notes in Computer Science, vol. 1423, pp. 267–288. Springer (1998)
24. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Res. **12**(3), 415–440 (1987)
25. Kannwischer, M.J., Krausz, M., Petri, R., Yang, S.: pqm4: Benchmarking NIST additional post-quantum signature schemes on microcontrollers. IACR Cryptol. ePrint Arch. p. 112 (2024), https://eprint.iacr.org/2024/112
26. Kim, J., Park, J.H.: NTRU++: compact construction of NTRU using simple encoding method. IEEE Trans. Inf. Forensics Secur. **18**, 4760–4774 (2023)
27. Kirchner, P., Fouque, P.: An improved BKW algorithm for LWE with applications to cryptography and lattices. In: CRYPTO (1). Lecture Notes in Computer Science, vol. 9215, pp. 43–62. Springer (2015)
28. Liang, Z., Fang, B., Zheng, J., Zhao, Y.: Compact and efficient kems over NTRU lattices. Comput. Stand. Interfaces **89**, 103828 (2024)
29. Liu, S., Sakzad, A.: Lattice codes for crystals-kyber. Des. Codes Cryptogr. **93**(8), 3181–3205 (2025)
30. Liu, Y., Zhang, Y., Lu, X., Cheng, Y., Yin, Y.: DAWN: smaller and faster NTRU encryption via double encoding. IACR Cryptol. ePrint Arch. p. 1520 (2025)
31. Lyubashevsky, V., Seiler, G.: NTTRU: truly fast NTRU using NTT. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2019**(3), 180–201 (2019)

32. MATZOV: Report on the security of LWE: Improved dual lattice attack. report (2022), https://api.semanticscholar.org/CorpusID:251600824
33. Paiva, T.B., Jr., M.A.S., Hafiz, S.M., Yildiz, B., Cominetti, E.L., Ogawa, H.S.: Tailorable codes for lattice-based kems with applications to compact ML-KEM instantiations. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2025**(3), 139–163 (2025)
34. Schwabe, P., Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Seiler, G., Stehlé, D.: CRYSTALS-KYBER. Tech. rep., National Institute of Standards and Technology (2020), available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions
35. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)
36. Wang, A., Zheng, Z., Zhao, C., Qiu, Z., Zeng, G., Yuan, Y., Mu, C., Wang, X.: Scloud^{+}: An efficient lwe-based KEM without ring/module structure. In: SSR. pp. 147–174. Lecture Notes in Computer Science, Springer (2024)
37. Zhang, J., Feng, D., Yan, D.: NEV: faster and smaller NTRU encryption using vector decoding. In: ASIACRYPT (7). Lecture Notes in Computer Science, vol. 14444, pp. 157–189. Springer (2023)
38. Zhou, G., Wang, L., Cao, Y., Huang, B., Hou, S., Gu, D.: Refined evaluation methods of decryption failure rate in lattice-based public-key encryption with message encoding. Cryptology ePrint Archive, Paper 2026/1350 (2026), https://eprint.iacr.org/2026/1350

Appendix A Benchmarks and Comparisons

We benchmark our schemes and related works on a 64-bit Ubuntu Linux system with an Intel i9-14900K CPU and 64 GB memory, reporting average CPU cycles over 100000 runs in Table 3. All implementations use identical optimization flags "-march=native -mtune=native -O3 -fomit-frame-pointer". The table groups results by security levels, where "Speedup" denotes the ratio of total Ref/AVX2 cycles for ephemeral key exchange relative to the first scheme in each group. We did not include DAWN and NEV because they are not open source. Compared to the similarly structured NTRU lattice-based constructions NTRU-HPS and NTRU-HRSS, the Ref implementation shows an improvement of 8.76 to 15.69 times, while the AVX2 implementation achieves an enhancement of 3.84 to 6.28 times. Additionally, when compared to SNTRU-Prime, the DTRU-Prime Ref implementation improves performance by 30.85 times. Compared to Kyber, our solution achieves a performance improvement of 1.05 to 1.32 times in the Ref implementation, and 1.10 to 1.22 times in the AVX2 implementation. Therefore, compared to alternatives with the same security level, our scheme demonstrates superior performance.

We also benchmark on the STM32F407 discovery board with ARMv7E-M architecture, featuring 196KiB RAM, 1MiB flash memory, and a 24MHz clock frequency. The compiling toolchain is arm-none-eabi-gcc version 13.2 with the -O3 option. Our benchmarking setup is based on pqm4 [25], ensuring fair comparisons by running all algorithms under identical conditions. Results are shown

Table 3: Comparison between DTRU, NTRU KEMS and Kyber in efficiency

Security Level	Scheme	Ref (CPU Cycles) KeyGen	Encap	Decap	AVX2 (CPU Cycles) KeyGen	Encap	Decap	Speedup
128 bits	**DTRU-512**	52111	36041	74447	15577	8215	18039	-
	NTRU-HPS$^{677}_{2048}$	1529764	88441	133453	127971	27510	21774	10.77×/4.24×
	NTRU-HRSS-701	1577211	56509	143098	119782	17319	23328	10.93× /3.84×
	Kyber-512	51108	55278	74056	11194	11719	12496	1.11×/0.85×
	DTRU-Light	43089	30061	63806	11299	4622	13096	-
	Kyber-512	51108	55278	74056	11194	11719	12496	1.32×/1.22×
192 bits	**DTRU-768**	95861	55848	116763	27526	8729	22868	-
	NTRU-HPS$^{821}_{4096}$	2097221	101614	152448	311213	30919	29361	8.76×/6.28×
	Kyber-768	83852	87783	111414	18557	18023	19349	1.05×/0.95×
256 bits	**DTRU-1024**	135695	78604	167703	21490	14010	37400	-
	NTRU-HPS$^{1229}_{4096}$	4021970	163655	231993	-	-	-	11.56×/-
	NTRU-HRSS-1373	5529595	127923	335902	-	-	-	15.69×/-
	Kyber-1024	129076	127675	157109	25893	25908	28088	1.08×/1.10×
384 bits	**DTRU-1536**	167081	121843	260110	23406	18627	54013	-
512 bits	**DTRU-2048**	341882	175062	375709	42304	27280	74396	-
256 bits (LPPNF)	**DTRU-Prime**	9703538	92796	195949	493544	32756	77594	-
	SNTRU-Prime-1277	188274036	30558130	89379889	-	-	-	30.85×/-

in Table 4. The meaning of "Speedup" is the same as in Table 3. It can be observed that on the ARM platform, our solution also demonstrates superior performance, achieving a performance improvement of 1.06 to 1.62 times compared to Kyber at the same security level. In summary, our solution exhibits high superiority across all platforms.

Table 4: Comparison between DTRU and Kyber in efficiency (ARM Cortex-M4)

Security Level	Scheme	ARM (CPU Cycles) KeyGen	Encap	Decap	Speedup
128 bits	**DTRU-512**	370574	259532	525733	-
	DTRU-Light	201970	144491	349529	-
	Kyber-512	394963	393752	431967	1.06×
192 bits	**DTRU-768**	370685	285777	632473	-
	Kyber-768	643632	660817	711204	1.56×
256 bits	**DTRU-1024**	625050	403312	919740	-
	Kyber-1024	1021507	1034854	1099259	1.62×
384 bits	**DTRU-1536**	834886	477772	1116494	-
512 bits	**DTRU-2048**	1746327	1300214	2889820	-
256 bits(LPPNF)	**DTRU-Prime**	238344450	561443	1320033	-

Appendix B Proof of Theorem 2

Proof. Let $\mathcal{A}$ be an adversary against the IND-CPA security of DTRU.PKE. We prove the theorem via a sequence of games G_0, G_1, G_2, where G_0 is the original IND-CPA game and G_2 is the random guess game. Let S_i be the event that $\mathcal{A}$ wins in game G_i.

Game G_0: The original IND-CPA game. We have $\mathrm{Adv}^{\text{IND-CPA}}_{\text{DTRU.PKE}}(\mathcal{A}) = \left|\Pr[S_0] - \frac{1}{2}\right|$.

- **Key Generation.** The challenger generates $(h = g/f, f)$ by running the KeyGen algorithm, and sends h to $\mathcal{A}$.
- **Query.** The adversary $\mathcal{A}$ is allowed to make adaptive encryption queries to the challenger, who responds by running the Enc algorithm.
- **Challenge.** The adversary $\mathcal{A}$ submits two messages $m_0, m_1 \in \mathcal{M}$ to the challenger. The challenger chooses a random bit $b \in \{0, 1\}$, computes the ciphertext $c^* = \mathrm{Enc}(pk, m_b)$ and sends it to $\mathcal{A}$. The adversary $\mathcal{A}$ outputs a guess b' of b and wins if $b' = b$.

Game G_1: Same as G_0 but h is uniformly random in R_q instead of g/f. Any distinguisher between G_0 and G_1 breaks the Decisional NTRU assumption:

$$|\Pr[S_0] - \Pr[S_1]| \leq \mathrm{Adv}^{\text{NTRU}}_{\Psi_f, \Psi_g}(\mathcal{B}).$$

Game G_2: Same as G_1 but $hr + e$ is replaced by uniform $u \in R_q$. Any distinguisher between G_1 and G_2 breaks the RLWE assumption:

$$|\Pr[S_1] - \Pr[S_2]| \leq \mathrm{Adv}^{\text{RLWE}}_{\Psi_r, \Psi_e}(\mathcal{C}).$$

In G_2, u is uniform, so c is independent of m and $\Pr[S_2] = 1/2$. Combining:

$$\begin{aligned}\mathrm{Adv}^{\text{IND-CPA}}_{\text{DTRU.PKE}}(\mathcal{A}) = \left|\Pr[S_0] - \frac{1}{2}\right| &\leq |\Pr[S_0] - \Pr[S_1]| + |\Pr[S_1] - \Pr[S_2]| \\ &\leq \mathrm{Adv}^{\text{NTRU}}_{\Psi_f, \Psi_g}(\mathcal{B}) + \mathrm{Adv}^{\text{RLWE}}_{\Psi_r, \Psi_e}(\mathcal{C}).\end{aligned}$$