DAWN: Smaller and Faster NTRU Encryption via Double Encoding

Yijian Liu[1,2], Yu Zhang[1,2], Xianhui Lu[1,2](✉), Yao Cheng[1,2], and Yongjian Yin[1,2]

[1] State Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{liuyijian, luxianhui}@iie.ac.cn

Abstract. This paper introduces DAWN, a compact and efficient NTRU encryption utilizing double encoding, which is provably secure under the NTRU assumption and the Ring-LWE assumption. We propose a technique for NTRU encryption called the zero divisor encoding. Unlike the polynomial encoding technique proposed by Hoffstein and Silverman (2001) and the vector encoding technique proposed by Zhang, Feng, and Yan in NEV (Asiacrypt 2023), our zero divisor encoding technique leverages the algebraic structure of the ring used in NTRU, enabling greater ciphertext compression while maintaining negligible decryption failure.

We further develop a paradigm for NTRU encryption called the double encoding paradigm to maximize the potential of the zero divisor encoding. This paradigm transforms optimizing an NTRU-based encryption into constructing a better encoding within the NTRU context, providing more concrete direction for scheme development. Several previous NTRU encryptions can be situated within this paradigm with different parameters, facilitating direct comparison. We instantiate this paradigm based on the provably IND-CPA secure NTRU variant by Stehlé and Steinfeld (Eurocrypt 2011) to achieve an IND-CPA secure PKE, and subsequently employ the Fujisaki-Okamoto transformation to achieve an IND-CCA secure KEM.

We present two parameter settings of DAWN: DAWN-α minimizes ciphertext size, achieving lengths of 436 bytes under NIST-I security and 973 bytes under NIST-V security; DAWN-β minimizes the combined size of the public key and ciphertext, attaining combined sizes of 964 bytes under NIST-I security and 2054 bytes under NIST-V security. DAWN achieves superior compactness and performance among current lattice-based KEMs without introducing additional security assumptions. Compared to NEV (Asiacrypt 2023), the previously leading NTRU-based KEM in balancing compactness and performance, DAWN demonstrates 20%–29% greater compactness at approximate security levels and decryption failure probabilities, while executing 1.1X–2.0X faster in a complete ephemeral key exchange process.

Keywords: Lattice based cryptography· NTRU · PKE · KEM

1 Introduction

NTRU, originally proposed by Hoffstein, Pipher, and Silverman [31], represents one of the earliest lattice-based encryption schemes. Although not initially conceptualized about lattices, extensive research has established that NTRU's security foundation rests on two principal hardness assumptions: the decisional NTRU assumption and the decisional Ring-Learning With Errors (Ring-LWE) assumption. Both assumptions are defined over the quotient ring $\mathcal{R}_{\boldsymbol{\phi}} = \mathbb{Z}[x]/\boldsymbol{\phi}$ and are fundamentally connected to the Shortest Vector Problem (SVP) in ideal lattices.

The decisional NTRU assumption posits that polynomial distributions of the form $(\boldsymbol{g}\boldsymbol{f}^{-1}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$ are computationally indistinguishable from the uniform distribution over $\mathcal{R}_{\boldsymbol{\phi},q} = \mathbb{Z}_q[x]/\boldsymbol{\phi}$, where $\boldsymbol{g}$ and $\boldsymbol{f}$ are polynomials with small coefficients. Complementing this, the decisional Ring-LWE assumption asserts that distributions of the form $(\boldsymbol{a}\boldsymbol{s} + \boldsymbol{e}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$ are computationally indistinguishable from the uniform distribution over $\mathcal{R}_{\boldsymbol{\phi},q}$, where $\boldsymbol{a}$ is uniformly sampled from $\mathcal{R}_{\boldsymbol{\phi},q}$, while $\boldsymbol{s}$ and $\boldsymbol{e}$ are polynomials with small coefficients. These robust hardness assumptions underpin the post-quantum security characteristics of NTRU-based schemes.

During the NIST post-quantum cryptography (PQC) standardization process [45], two NTRU variants demonstrated significant promise: NTRU Prime [12] advanced to the third round of candidates, while NTRU [21] (originating from [31], a merger of NTRU-HPS and NTRU-HRSS) progressed to the third round of finalists. Despite their advancement, neither scheme achieved final standardization, primarily because they failed to outperform Kyber [14], the eventual selection, in terms of efficiency or compactness at equivalent security levels [1]. Nevertheless, the cryptographic community continues to actively explore and enhance NTRU structures, with numerous improvements and variants proposed in recent years [43,37,28,24,51,11,39], significantly enhancing the competitiveness and efficacy of NTRU-based schemes.

The evolution of NTRU's underlying algebraic structure reveals an ongoing quest for optimal balance between security and performance. The original NTRU implementation utilized the ring $\mathcal{R}_{x^n-1,q}$, where q is a power of 2 and n is prime. NTRU Prime [12] modified by adopting $\mathcal{R}_{x^n-x-1,q}$ to enhance resistance against potential algebraic attacks. However, these two ring structures do not naturally support the Number Theoretic Transform (NTT), even though techniques such as NTT-unfriendly [20] and Toom-Cook [6] can be implemented, resulting in performance limitations.

A breakthrough came with Lyubashevsky and Seiler's introduction of NTTRU [43], which employed the specific ring $\mathcal{R}_{x^{768}-x^{384}+1,7681}$ that naturally supports NTT operations. Building on this approach, Duman *et al.* [24] extended the concept to general tri-cyclotomic rings of the form $\mathcal{R}_{x^n-x^{n/2}+1,q}$, where q is prime and n is a product of powers of 2 and 3, thereby offering more flexible parameter configurations. NTRU+ [37] similarly adopts this tri-cyclotomic ring structure. Furthermore, CNTR and CTRU [39] employed the E_8 lattice code to encode messages for enhanced compression tolerance; these schemes also utilize parameters over such tri-cyclotomic rings and achieve excellent compactness and efficiency, particularly at the NIST-III security level. A challenge with these rings, however, is that the norms of vectors in the anti-cyclic matrix corresponding to the polynomial are different, complicating security analysis and decryption error rate calculations.

Drawing inspiration from the NTRU-based signature scheme Falcon [27], Fouque *et al.* proposed BAT [28], which reformulates the decryption process as a Bounded Distance Decoding problem and resolves it using the lattice trapdoor created by $(\boldsymbol{g}, \boldsymbol{f})$. BAT uses the cyclotomic ring $\mathcal{R}_{x^n+1,q}$ where n is a power of 2, enhances performance by establishing fixed precision and substituting floating-point computations with integer operations through the NTRU structure. Additionally, BAT achieves significant ciphertext compression by replacing the Ring-LWE assumption with the Ring Learning With Rounding (Ring-LWR) assumption. While BAT currently exhibits the most compact public key and ciphertext sizes among all current NTRU-based KEM schemes, it incurs substantial key generation costs due to the computation of the trapdoor.

Departing from conventional NTRU-based encryption schemes that employ low-part encoding decryption processes, Zhang, Feng, and Yan introduced NEV [51], which operates over the same ring as BAT. NEV employs a vector encoding technique, resulting in a decryption process that resembles high-part encoding with a repetition code, thereby enhancing decryption capability. NEV achieves an optimal balance between size and speed without requiring a trapdoor, delivering smaller public key and ciphertext sizes coupled with superior performance compared to Kyber [14]. In a parallel development, Chitchanok *et al.* proposed a variant of Falcon based on module-NTRU [19], and subsequently Bai *et al.* proposed a KEM based on module-NTRU [11], which, while slightly smaller than NEV in terms of ciphertext, represents another promising direction in NTRU research.

1.1 Our Results

We present DAWN[1].PKE, a compact and efficient NTRU-based encryption scheme that achieves provable IND-CPA security under the NTRU assumption and the Ring-LWE assumption over the power-of-2 cyclotomic ring $\mathcal{R}_{x^n+1}$ in the quantum random oracle model. We then obtain an IND-CCA secure DAWN.KEM by applying the standard Fujisaki-Okamoto transformation to DAWN.PKE. Our first contribution is a new encoding technique—*zero divisor encoding*—which exploits the algebraic features of the quotient ring used in NTRU. Furthermore, we introduce a generic paradigm for NTRU encryption—*double encoding paradigm*—which transforms the challenge of designing more compact and efficient NTRU encryption schemes into constructing superior algebraic coding schemes specifically tailored for the NTRU context. This paradigm complements the zero divisor encoding exceptionally well, maximizing its capabilities. By providing a clearer direction, this paradigm enables us to optimize the NTRU-based

[1] DAWN stands for "Double encoding Applied With NTRU encryption"

encryption methodically. Leveraging this paradigm, we have developed the smallest and fastest NTRU-based KEM to date without requiring additional security assumptions.

We present two parameter settings designed for different scenarios:

- DAWN-α: Optimized to minimize ciphertext size, making it suitable for scenarios involving a single key generation followed by multiple encapsulations and decapsulations, such as secure messaging systems.
- DAWN-β: Optimized to minimize the overall communication overhead (the combined size of public key and ciphertext), making it suitable for applications where both components carry equal importance, such as key exchange protocols.

Both variants employ small moduli—769 for DAWN-α and 257 for DAWN-β—and utilize rounding computations for ciphertext compression. Importantly, unlike schemes that rely on the Ring-LWR assumption (e.g., BAT [28]), DAWN's security does not depend on quantifying the compression-induced noise. In this respect, DAWN follows Kyber's more conservative approach to security estimation, employing ciphertext compression without incorporating the resulting noise into security estimates, thereby providing more robust security guarantees.

As demonstrated in Table 1, DAWN exhibits significant storage efficiency advantages compared to state-of-the-art alternatives. Under the same security level, DAWN-α ciphertexts are 20%–29% more compact than NEV and 3%–8% more compact than BAT. Similarly, DAWN-β achieves 16%–21% greater compactness than NEV and 3%–8% greater compactness than BAT. The improvements over NIST-standardized Kyber are even more substantial: DAWN-α ciphertexts are 38%–43% smaller, while DAWN-β reduces overall communication overhead by 34%–38%. (see Table 1)

Beyond size efficiency, DAWN demonstrates superior computational performance, as detailed in Table 2. When measuring the combined computational costs of key generation, encapsulation, and decapsulation operations, DAWN outperforms BAT by factors of 167.7X to 1386.6X. Compared to other efficient schemes, DAWN maintains its advantage, operating 1.1X to 2.0X faster than NEV and 1.5X to 2.9X faster than Kyber. These performance metrics establish DAWN as the leading NTRU-based KEM in terms of both storage efficiency and computational performance. (see Table 2)

A distinctive strength of DAWN is its exceptional resilience to decryption failures, with the plaintext term exerting minimal influence on failure probability. This characteristic ensures a narrow gap between worst-case and average-case decryption failure probabilities, providing robust protection against chosen-ciphertext attacks that typically exploit such disparities (e.g., [25][30]).

While DAWN does not feature a dedicated NIST level III parameter set, our NIST level V implementations demonstrate remarkable efficiency relative to NIST level III schemes. DAWN's implementations yield smaller overall sizes than other NIST level III schemes, except only marginally larger than MNTRU-II(b), which has yet to be implemented. This cross-level comparison further underscores DAWN's exceptional efficiency, achieving higher security with comparable or better resource requirements than lower-security alternatives. (see Table 1)

1.2 Our Techniques

We start with a general and simple form of NTRU encryption. Let q, p be two coprime positive integers, and $\mathcal{R}_{\boldsymbol{\phi},q}$ be the quotient ring $\mathbb{Z}_q[x]/\boldsymbol{\phi}$. We calculate the public key $\boldsymbol{h} \equiv (\boldsymbol{g}\boldsymbol{f}^{-1}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$, and for a message polynomial $\boldsymbol{m}$, we encrypt it to the ciphertext $\boldsymbol{c} \equiv (\boldsymbol{h}\boldsymbol{s} + \boldsymbol{e} + p^{-1}\boldsymbol{m}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$. Some schemes directly embed $\boldsymbol{m}$ in $\boldsymbol{e}$; however, due to the necessity of the reduction to Ring-LWE assumption, this approach constrains the distribution of $\boldsymbol{m}$ and potentially results in a weaker reduction factor, as demonstrated in NEV′ [51] under the Subset-Sum Parity Ring-LWE (ssp Ring-LWE) assumption. For decryption, we calculate $\boldsymbol{m}' \equiv (p \cdot \boldsymbol{c}\boldsymbol{f}) \equiv (p \cdot (\boldsymbol{g}\boldsymbol{s} + \boldsymbol{f}\boldsymbol{e}) + \boldsymbol{f}\boldsymbol{m}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$, which can be transformed from $\mathbb{Z}_q[x]$ into $\mathbb{Z}[x]$ if $\|p \cdot (\boldsymbol{g}\boldsymbol{s} + \boldsymbol{f}\boldsymbol{e}) + \boldsymbol{f}\boldsymbol{m}\|_\infty < \frac{q}{2}$, representing the condition for successful decryption. The more relaxed this condition, the better the decryption capability, yielding greater compactness under equivalent security levels. Numerous prior NTRU-based schemes (e.g., [12,31,34,43]) select p as 2 or 3 to minimize the noise amplification factor.

Our first observation is that we can replace the integer p with a polynomial $\boldsymbol{t}$ that provides a lower amplification factor. This approach differs fundamentally from the polynomial encoding proposed in [32], which uses p as the polynomial $x + 2$, requiring it to be coprime with $\boldsymbol{\phi}$ under the ring $\mathbb{Z}_q[x]$ while maintaining sufficient message space. Instead, we analyze the core reasoning behind NTRU decryption: the successful decryption condition ensures that the equation holds on the quotient ring $\mathbb{Z}[x]/\boldsymbol{\phi}$, which then transforms into a quotient ring $\mathbb{Z}_p[x]/\boldsymbol{\phi}$ to eliminate noise terms and reveal the message. This mechanism functions effectively because p equals zero in this quotient ring.

We identify alternative zero factors that can perform the same function, specifically $x^{n/2} + 1$ for the quotient ring $\mathbb{Z}_2[x]/(x^n + 1)$ where n is a power of 2. This zero factor eliminates noise terms during decryption while only reducing the message space size. The critical advantage lies in the amplification factor: $x^{n/2} + 1$ corresponds to $\sqrt{2}$, whereas the smallest amplification achievable with the polynomial encoding in [32] is $\sqrt{5}$, and traditional approaches in most NTRU-based encryptions use factors of 2 or 3. We term this technique *zero factor encoding*, distinguishing it from the conventional approach that functions as a p low-bit encoding.

From a high-level perspective, the underlying principle behind the polynomial encoding technique in [32], the vector encoding technique in [51], and our zero divisor encoding technique all leverage redundant message space to increase error tolerance. Specifically, a typical selection of the degree n for 128-bit security based on the NTRU assumption is 512, providing up to 384 redundant bits to exploit. While BAT [28] follows a technical route similar to Falcon [27] that reduces p to 1 using a trapdoor, the computational expense of generating this trapdoor makes its key generation prohibitively time-consuming.

Based on the zero factor encoding, we propose a new generic paradigm called the double encoding paradigm for NTRU-based encryption to maximize the potential of this technique. For the polynomial $\boldsymbol{\phi}$ and two primes q, p, we select an l_t-degree polynomial $\boldsymbol{t}$ such that $\boldsymbol{\phi} \equiv 0 \bmod (\boldsymbol{t}, \mathbb{Z}_p)$. For a message polynomial $\boldsymbol{m}$ with at most l_m-degree, we choose an l_w-degree polynomial $\boldsymbol{w}$ with the capacity to correct errors in the set $\mathcal{C}_{\boldsymbol{w}}$, with degrees satisfying $l_w + l_m \leq l_t$. These polynomials $\boldsymbol{t}$ and $\boldsymbol{w}$ function as two complementary encoding layers, which is the source of the term "double encoding".

The double encoding paradigm operates as follows: We use the public key $\boldsymbol{h} \equiv (\boldsymbol{g}\boldsymbol{f}^{-1}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$ and encrypt the message polynomial $\boldsymbol{m}$ to obtain ciphertext $\boldsymbol{c} \equiv \big(\boldsymbol{h}\boldsymbol{s} + \boldsymbol{e} + \boldsymbol{t}^{-1}\boldsymbol{w}\boldsymbol{m}\big) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$. For decryption, we first calculate $\boldsymbol{c}' \equiv (\boldsymbol{c}\boldsymbol{t}\boldsymbol{f}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$. Then, leveraging polynomial $\boldsymbol{t}$, we compute $\boldsymbol{m}' \equiv (\boldsymbol{c}'\boldsymbol{f}^{-1}) \bmod (\boldsymbol{t}, \mathbb{Z}_p)$ to eliminate the term $\boldsymbol{t}(\boldsymbol{g}\boldsymbol{s}+\boldsymbol{f}\boldsymbol{e})$. Furthermore, letting $\boldsymbol{e}' \triangleq \lfloor \frac{1}{q} \cdot (\boldsymbol{t}(\boldsymbol{g}\boldsymbol{s} + \boldsymbol{f}\boldsymbol{e}) + \boldsymbol{f}\boldsymbol{w}\boldsymbol{m}) \rceil$, we derive $\boldsymbol{m}' \equiv (\boldsymbol{w}\boldsymbol{m} - q \cdot \boldsymbol{e}'\boldsymbol{f}^{-1}) \bmod (\boldsymbol{t}, \mathbb{Z}_p)$. Finally, we employ the decoding algorithm corresponding to $\boldsymbol{w}$ to extract $\boldsymbol{m}$ from $\boldsymbol{m}'$.

Under this paradigm, the successful decryption condition is $\boldsymbol{e}' \in \mathcal{C}_{\boldsymbol{w}}$. Even in cases where $\boldsymbol{w}$ cannot correct errors (i.e., $\boldsymbol{w} = 1, \mathcal{C}_{\boldsymbol{w}} = \{0\}$), this condition remains more favorable than the original NTRU decryption condition and is comparable to NEV. We select the quotient ring $\mathbb{Z}[x]/(x^n + 1)$ where n is a power of 2, and choose $p = 2, \boldsymbol{t} = x^{n/2} + 1, \boldsymbol{w} = x^{n/4} + 1$, allowing $\boldsymbol{w}$ to correct one error with negligible computational overhead. In this configuration, given that $\boldsymbol{t} \equiv 0 \bmod (\boldsymbol{w}, \mathbb{Z}_p)$, we can directly obtain $(\boldsymbol{e}') \bmod (\boldsymbol{w}, \mathbb{Z}_p)$ by calculating $(\boldsymbol{c}') \bmod (\boldsymbol{w}, \mathbb{Z}_p)$, and then extract the complete message $\boldsymbol{m}$ with several computations over $\mathbb{Z}_q$. We provide a comparison of successful decryption conditions across different NTRU-based schemes to show our advantages in Table 4 and Figure 1. This comparison of decryption conditions elucidates our advantages among all NTRU-based encryption schemes: firstly, our error tolerance is superior without requiring stronger hardness assumptions; secondly, schemes (e.g., BAT, NEV′) that employ stronger hardness assumptions must utilize more constrained polynomial distributions or rounding computations compared to DAWN. These factors contribute to the leadership of our scheme.

For key generation in DAWN, we must calculate $(\boldsymbol{f}^{-1}) \bmod (\boldsymbol{t}, \mathbb{Z}_p)$. Since Number Theoretic Transform (NTT) is not applicable for this calculation, we leverage Hensel's lifting lemma and introduce an algorithm that exploits the identity $x^{2^l} + 1 = (x + 1)^{2^l}$ over $\mathbb{Z}_2$ to compute this inverse efficiently.

A particularly advantageous feature of DAWN is the minimal impact that message selection has on decryption failure probability. Our analysis demonstrates that DAWN achieves average-case decryption failure probabilities of 2^{-133} and 2^{-163} (2^{-130} and 2^{-138}) under NIST security levels I and V. Even in worst-case scenarios where an adversary selects ciphertexts corresponding to the most challenging message consisting entirely of 1, these probabilities remain remarkably low at 2^{-130} and 2^{-156} (2^{-119} and 2^{-114}), with gap factors ranging from 2^3 to 2^7 (2^{11} to 2^{24}). For comparison, NTRU-A [24] exhibits gap factors exceeding 2^{100}, while NEV [51] demonstrates gap factors approximately comparable to DAWN. Consequently, following the strategy implemented in Kyber and NEV, we employ a Fujisaki-Okamoto transformation resistant to multi-target attacks, which, combined with the aforementioned features, enables DAWN to effectively withstand potential CCA attacks such as those described in [30].

1.3 Comparison to the State of the Art

DAWN achieves the smallest combined ciphertext and public key sizes among all known lattice-based KEMs. Before our work, BAT and NEV offered the most competitive size-performance trade-offs in this domain. Tables 1, 2, 3, and 4 present comprehensive comparisons between DAWN and leading lattice-based candidates, including mainstream NTRU-based KEMs and Kyber. We compare our scheme against BAT [28], MNTRU [11], NEV and NEV′ [51], CNTR and CTRU [39], NTRU-A [24], NTRU-HPS and NTRU-HRSS [21], NTTRU [43], and Kyber [14].

For fair comparison, we propose a variant called DAWN-α-512 of DAWN-α at the NIST-I security level, which uses the same plaintext length and shared key length (256 bits) as Kyber and NEV. We choose the parameters $\boldsymbol{t} = x^{n/2} + 1$ and $\boldsymbol{w} = 1$ for this variant, which essentially degenerates into "single encoding" and results in slightly reduced efficiency and compactness.

For security assessment, we employ the widely adopted "lattice estimator" tool [5], maintaining consistency with the methodology used in contemporary lattice-based proposals. To ensure equitable comparison, we use the recent version (up to commit 5ba00f5) to evaluate the security levels of all schemes.

For compactness evaluation, we focus primarily on two key metrics: the size of the ciphertext and the combined size of the public key and ciphertext. Due to different design motivations, we compare the ciphertext size with DAWN-α and the overhead size with DAWN-β. As demonstrated in Table 1, DAWN-α consistently achieves the smallest ciphertext sizes across comparable security levels, reducing size requirements by 3%–65% relative to alternative schemes. Similarly, DAWN-β consistently achieves the smallest overhead sizes across comparable security levels, reducing size requirements by 3%–61% relative to alternative schemes.

For performance evaluation, we assess the total computational cost of a complete key generation, encapsulation, and decapsulation cycle. As detailed in Table 2, DAWN demonstrates superior overall efficiency, consistently outperforming competing KEMs by factors ranging from 1.1X to 35.2X. To ensure methodological consistency and fair comparison, all schemes in Table 2 utilize SHA3 and SHAKE256 for hashing and pseudorandom generation functions. Since BAT only provides implementations with BLAKE-based primitives, we make a separate comparison between BAT and DAWN with BLAKE primitives in Table 3. All performance measurements were conducted under controlled conditions on identical hardware platforms, utilizing both reference C implementations and optimized AVX2 implementations. Table 10 provides more detailed timing data of DAWN.

Table 1: Comparison between DAWN, NTRU KEMs, and Kyber in size

Scheme	NIST Sec.	Ciphertext (bytes)	Public Key (bytes)	Sec. Level ($\log_2$)	Dec. Failure ($\log_2$)	Compression Ratio (α/β)
DAWN-α-512	NIST-I	436	615	140	-133	-/-
DAWN-α-512′		514	615	140	-166	-/-
DAWN-β-512		450	514	134	-130	-/-
BAT-512		473	521	144	-146	8%/3%
MNTRU-I		614	646	142	-131	29%/23%
NEV-512		615	615	137	-138	29%/21%
NEV′-512		615	615	142	-200	29%/21%
CNTR-512		640	768	148	-170	32%/32%
CTRU-512		640	768	139	-143	32%/32%
NTRU-A$^{576}_{2593}$		864	864	144	-150	50%/44%
NTRU-HPS$^{677}_{2048}$		930	930	165	$-\infty$	53%/48%
NTRU-HRSS-701		1138	1138	156	$-\infty$	62%/57%
NTTRU-768		1248	1248	170	-1217	65%/61%
Kyber-512		768	800	140	-138	43%/38%
MNTRU-II(b)	NIST-III	921	953	201	-129	$-5\%/-9\%$
CNTR-768		960	1152	206	-230	$-1\%/3\%$
CTRU-768		960	1152	195	-184	$-1\%/3\%$
NTRU-A$^{768}_{3457}$		1152	1152	195	-202	15%/10%
NTRU-HPS$^{821}_{4096}$		1230	1230	193	$-\infty$	20%/16%
Kyber-768		1088	1184	196	-164	10%/9%
DAWN-α-1024	NIST-V	973	1229	270	-163	-/-
DAWN-β-1024		1027	1027	267	-138	-/-
BAT-1024		1006	1230	272	-166	3%/8%
MNTRU-III(b)		1228	1260	267	-131	20%/17%
NEV-1024		1229	1229	267	-152	20%/16%
NEV′-1024		1229	1229	279	-200	20%/16%
CNTR-1024		1280	1536	261	-291	24%/27%
CTRU-1024		1408	1536	261	-195	24%/30%
NTRU-A$^{1152}_{3457}$		1728	1728	294	-140	43%/40%
NTRU-HPS$^{1229}_{4096}$		1842	1842	282	$-\infty$	47%/44%
NTRU-HRSS-1373		2401	2401	289	$-\infty$	59%/57%
Kyber-1024		1568	1568	262	-174	38%/34%

We also provide a comparison of the successful decryption conditions and the hardness assumptions in Table 4. Due to the difference between module and ring structures, it is challenging to directly compare the successful decryption condition between MNTRU and other NTRU-based KEMs. For simplicity, we omit the polynomials in the form of $x^i + 1$, and replace them with a factor $\sqrt{2}$, which affects all schemes equally in the calculation of decryption failure probability. Although BAT incorporates two distinct successful decryption conditions, its parameters are specifically calibrated to align these conditions; therefore, we present only one condition for clarity. For NEV and NEV′, we set the parameter k to 4, which corresponds to their optimal decryption condition scenario.

Table 2: Comparison between DAWN, NTRU KEMs, and Kyber in performance

Scheme	NIST Sec.	REF				AVX2			
		Keygen (kcycles)	Encaps (kcycles)	Decaps (kcycles)	Speed-up Ratio (α/β)	Keygen (kcycles)	Encaps (kcycles)	Decaps (kcycles)	Speed-up Ratio (α/β)
DAWN-α-512	NIST-I	92	42	45	-/-	22	20	19	-/-
DAWN-α-512$'$		107	44	49	-/-	34	22	23	-/-
DAWN-β-512		71	35	48	-/-	23	22	23	-/-
*NEV-512		95	88	113	1.7X/1.9X	21	33	26	1.3X/1.2X
*NEV$'$-512		89	83	110	1.6X/1.8X	20	30	23	1.2X/1.1X
NTRU-HPS$^{677}_{2048}$		3, 095	181	269	19.8X/23.0X	249	60	53	5.9X/5.3X
NTRU-HRSS-701		3, 307	112	290	20.7X/24.1X	231	36	54	5.3X/4.7X
Kyber-512		118	131	134	2.1X/2.5X	39	41	24	1.7X/1.5X
*CNTR-768	NIST-III	118	65	133	0.9X/0.9X	13	11	35	0.5X/0.4X
*CTRU-768		118	63	135	0.9X/0.9X	11	12	36	0.5X/0.4X
NTRU-HPS$^{821}_{4096}$		3940	219	318	11.6X/11.9X	354	69	67	4.2X/3.6X
Kyber-768		188	208	238	1.6X/1.7X	53	56	37	1.2X/1.1X
DAWN-α-1024	NIST-V	138	77	113	-/-	46	36	38	-/-
DAWN-β-1024		135	88	127	-/-	46	42	47	-/-
*NEV-1024		208	183	257	2.0X/1.9X	37	64	50	1.3X/1.1X
*NEV$'$-1024		205	171	251	1.9X/1.8X	37	60	45	1.2X/1.1X
NTRU-HPS$^{1229}_{4096}$		8, 094	347	539	27.4X/25.7X	-	-	-	-/-
NTRU-HRSS-1373		10, 530	266	764	35.2X/33.0X	-	-	-	-/-
Kyber-1024		281	298	363	2.9X/2.7X	71	79	53	1.7X/1.5X

*: These schemes are not open source; we use the data from their papers [51], [39]

Table 3: Comparison between DAWN (BLAKE version) and BAT in performance

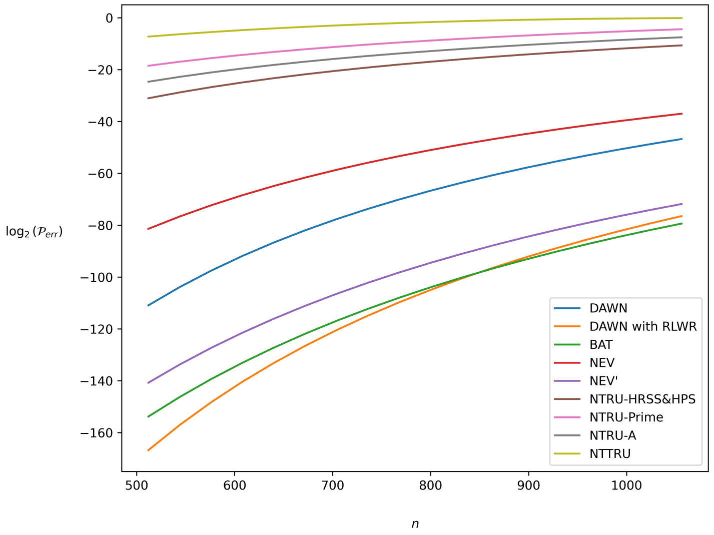
Scheme	NIST Sec.	REF				AVX2			
		Keygen (kcycles)	Encaps (kcycles)	Decaps (kcycles)	Speed-up Ratio (α/β)	Keygen (kcycles)	Encaps (kcycles)	Decaps (kcycles)	Speed-up Ratio (α/β)
DAWN-α-512	NIST-I	95	38	45	-/-	18	8	13	-/-
DAWN-β-512		83	32	49	-/-	19	11	17	-/-
BAT-512		29, 579	33	233	167.7X/182.0X	23, 543	9	50	605.2X/502.2X
DAWN-α-1024	NIST-V	147	69	115	-/-	37	15	27	-/-
DAWN-β-1024		144	87	134	-/-	39	22	37	-/-
BAT-1024		155, 582	65	444	471.6X/427.6X	109, 412	20	107	1386.6X/1117.7X

Figure 1 provides a comparative analysis of encryption schemes under standardized conditions. We established consistent parameters across all schemes by setting the modulus $q = 257$ and standardizing the coefficient distributions for $\boldsymbol{g}$, $\boldsymbol{f}$, $\boldsymbol{s}$, $\boldsymbol{e}$, and $\boldsymbol{m}$ as ternary distributions with standard deviation $\frac{1}{2}$. The x-axis represents dimension n, while the y-axis displays $\log_2(\mathcal{P}_{err})$, the logarithm of decryption failure probability. Our analysis demonstrates that DAWN achieves the lowest decryption failure probability among schemes that only require the Ring-LWE assumption. When implemented with the Ring-LWR assumption, DAWN exhibits the greatest error tolerance among all evaluated schemes. It is important to note that while certain schemes demonstrate lower decryption failures than standard DAWN in this specific metric, this does not translate to superior compactness in implementation. This distinction arises because NEV$'$ and BAT embed the message polynomial $\boldsymbol{m}$ directly into the error term $\boldsymbol{e}$, which fundamentally constrains their parameter selection flexibility. Furthermore, NEV′'s unique algebraic structure precludes the application of ciphertext compression through rounding operations, which explains the size disparities observed between DAWN, NEV′, and BAT. In conclusion, DAWN achieves the optimal balance of decryption reliability and implementation efficiency without requiring stronger hardness assumptions.

Table 4: Comparison among NTRU KEMs in conditions and assumptions

Scheme	Hardness Assumption	Decryption Condition
DAWN	NTRU + Ring-LWE $\mathcal{R}_{\boldsymbol{\phi},q} = \mathbb{Z}_q[x]/(x^n+1)$	$\lfloor \frac{\sqrt{2}}{q} \cdot (\boldsymbol{gs} + \boldsymbol{fe} + \boldsymbol{fm}) \rceil \in \{0\} \cup \{\pm x^i\}_{0 \le i < n}$
BAT	NTRU + Ring-LWR $\mathcal{R}_{\boldsymbol{\phi},q} = \mathbb{Z}_q[x]/(x^n+1)$	$\lfloor \frac{1}{q} \cdot (\boldsymbol{gs} + \boldsymbol{fe}) \rceil = 0$
NEV	NTRU + Ring-LWE $\mathcal{R}_{\boldsymbol{\phi},q} = \mathbb{Z}_q[x]/(x^n+1)$	$\lfloor \frac{1}{q-1} \cdot (\boldsymbol{gs} + \sqrt{2} \cdot \boldsymbol{fe} + \boldsymbol{fm} + \boldsymbol{e}) \rceil = 0$
NEV′	NTRU + ssp Ring-LWE $\mathcal{R}_{\boldsymbol{\phi},q} = \mathbb{Z}_q[x]/(x^n+1)$	$\lfloor \frac{1}{q-11} \cdot (\boldsymbol{gs} + \boldsymbol{fe}) \rceil = 0$
NTRU-HRSS&HPS	NTRU + Ring-LWE $\mathcal{R}_{\boldsymbol{\phi},q} = \mathbb{Z}_q[x]/(x^n-1)$	$\lfloor \frac{1}{q} \cdot (3 \cdot \boldsymbol{gs} + \boldsymbol{fe}) \rceil = 0$
NTRU-Prime	NTRU + Ring-LWE $\mathcal{R}_{\boldsymbol{\phi},q} = \mathbb{Z}_q[x]/(x^n-x-1)$	$\lfloor \frac{1}{q} \cdot (\boldsymbol{gs} + 3 \cdot \boldsymbol{fe}) \rceil = 0$
NTRU-A	NTRU + ssp Ring-LWE $\mathcal{R}_{\boldsymbol{\phi},q} = \mathbb{Z}_q[x]/(x^n-x^{n/2}+1)$	$\lfloor \frac{2}{q} \cdot (\boldsymbol{gs} + \boldsymbol{fe}) \rceil = 0$
NTTRU	NTRU + Ring-LWE $\mathcal{R}_{\boldsymbol{\phi},q} = \mathbb{Z}_q[x]/(x^n-x^{n/2}+1)$	$\lfloor \frac{3}{q} \cdot (\boldsymbol{gs} + \boldsymbol{fe}) \rceil = 0$

Fig. 1: Comparison among NTRU KEMs in decryption failure probabilities



2 Preliminaries

2.1 Notations

We use $\mathcal{R}_{\boldsymbol{\phi}}$ to denote the quotient ring $\mathbb{Z}[x]/\boldsymbol{\phi}$ and $\mathcal{R}_{\boldsymbol{\phi},q} = \mathcal{R}_{\boldsymbol{\phi}}/q\mathcal{R}_{\boldsymbol{\phi}}$ to denote the quotient ring $\mathbb{Z}_q[x]/\boldsymbol{\phi}$. We use $\mathbb{Z}_q$ to represent the set $\{-\frac{q-1}{2}, \ldots, 0, \ldots, \frac{q-1}{2}\}$, where q is a prime greater than 2. We denote by regular font letters (e.g. x, y) the element in $\mathbb{Z}$ or $\mathbb{Z}_q$, and by bold lower-case letters (e.g. $\boldsymbol{x}, \boldsymbol{y}$) the element in $\mathcal{R}_{\boldsymbol{\phi}}$ (resp. $\mathcal{R}_{\boldsymbol{\phi},q}$). We use $\bmod\,(\boldsymbol{\phi}, \mathcal{I})$ to represent the modular operation over $\mathcal{I}[x]/\boldsymbol{\phi}$, where $\mathcal{I}$ is a ring and $\boldsymbol{\phi}$ is a polynomial over $\mathcal{I}[x]$, e.g. $\bmod(x^n+1, \mathbb{Z}_2)$ denotes the modular operation over $\mathbb{Z}_2[x]/(x^n+1)$. For a polynomial $\boldsymbol{a}$, we denote the coefficient of x^i by $\boldsymbol{a}_{[i]}$, and we use $\overline{\boldsymbol{a}}$ to represent the conjugate of it such that $\overline{\boldsymbol{a}}(x) = \boldsymbol{a}(x^{-1})$.

Let $\lfloor a \rceil_p = \lfloor \frac{1}{p} \cdot a \rceil$ denote the p-rounding function, we will abuse the notion and write $\lfloor \boldsymbol{a} \rceil_p$ to mean a polynomial whose coefficients are $\lfloor \boldsymbol{a}_{[i]} \rceil_p$, we omit the index while $p = 1$. The notation $\mathcal{U}(\mathcal{V})$ represents the uniform distribution over the set $\mathcal{V}$, and the notation $\mathcal{T}_{n,k}$ denotes the ternary distribution, where the probability of 1 or -1 is $\frac{k}{n}$, and the probability of 0 is $\frac{n-2k}{n}$. We use the function $\mathbf{Sample}(\mathcal{X})$ to denote the sampling process of an element drawn from the distribution $\mathcal{X}$, and we abuse the notion $\mathbf{Sample}(\mathcal{X}, \rho)$ to denote the sampling process with the given random seed ρ. For a distribution $\mathcal{X}$, we denote by $\mathcal{X}^n$ the distribution of n-degree polynomials whose coefficients are drawn from $\mathcal{X}$. We say a function $\mu(\kappa)$ is negligible if $\mu \in o(\kappa^{-\omega(1)})$.

2.2 Public-Key Encryption

Definition 1 (Public-Key Encryption). *A Public-Key Encryption (PKE) scheme Π with message space $\mathcal{M}$ and the security parameter κ consists of three PPT algorithms* $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$*:*

- $\mathsf{KeyGen}(1^\kappa)$*: the input is the security parameter κ, and the output is the public key and secret key (pk, sk).*
- $\mathsf{Encrypt}(pk, m, \rho)$*: the inputs are the public key pk, a message $m \in \mathcal{M}$, and a randomness ρ, and the output is the ciphertext c. When the randomness is implicitly sampled within the algorithm, we abbreviate this as* $\mathsf{Encrypt}(pk, m)$*.*
- $\mathsf{Decrypt}(sk, c)$*: the inputs are the secret key sk and the ciphertext c, and the output is the message m.*

Correctness. The PKE scheme Π is *δ-correct* if

$$\mathbb{E}\left[\max_{m \in \mathcal{M}} \Pr[\mathsf{Decrypt}(sk, \mathsf{Encrypt}(pk, m)) \neq m]\right] \leq \delta,$$

where the expectation is over $(pk, sk) \leftarrow \mathsf{KeyGen}$ and the randomness of $\mathsf{Encrypt}$.

Definition 2 (IND-CPA). *A PKE scheme Π is called* IND-CPA *secure if for any PPT adversary $\mathcal{A}$, the advantage*

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

Fig. 2: **IND-CPA (PKE) and IND-CCA (KEM) Security Game**

```
$\mathsf{Game}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}}$:
$pk, sk \leftarrow \mathsf{KeyGen}(1^{\kappa})$
$(m_0, m_1) \leftarrow \mathcal{A}(pk)$
$b \leftarrow \{0,1\}$
$c_b \leftarrow \mathsf{Encrypt}(pk, m_b)$
$b' \leftarrow \mathcal{A}(c_b)$
```

```
$\mathsf{Game}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM},\mathcal{A}}$:
$(pk, sk) \leftarrow \mathsf{KeyGen}(1^{\kappa})$,
$(c^*, K_0^*) \leftarrow \mathsf{Encapsulation}(pk)$,
$K_1^* \xleftarrow{\$} \mathcal{K}$,
$b \xleftarrow{\$} \{0,1\}$,
$b' \leftarrow \mathcal{A}^{O_{\mathsf{Decaps}}(sk, \cdot \neq c^*)}(pk, c^*, K_b^*)$
```

in the $\mathsf{Game}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}}$ *in Fig. 2 is negligible in security parameter* κ.

2.3 Key Encapsulation Mechanism

Definition 3 (Key Encapsulation Mechanism). *A Key Encapsulation Mechanism (KEM) scheme with session key space* $\mathcal{K}$ *the security parameter* κ *consists of three PPT algorithms* ($\mathsf{KeyGen}, \mathsf{Encapsulation}, \mathsf{Decapsulation}$)*:*

- $\mathsf{KeyGen}(1^{\kappa})$*: the input is the security parameter* κ*, and the output is the public key and secret key* (pk, sk)*.*
- $\mathsf{Encapsulation}(pk)$ *: the inputs are the public key* pk*, and the output is a ciphertext* c *and a session key* $K \in \mathcal{K}$*.*
- $\mathsf{Decapsulation}(sk, c)$ *: the inputs are the ciphertext* c *and the secret key* sk*, and the output is the session key* $K \in \mathcal{K}$ *or a rejection symbol* $\perp$*.*

Correctness. The KEM scheme is called δ-correct if:

$$\mathbb{E}\left[\Pr[\mathsf{Decapsulation}(sk, c) \neq K, (c, K) \leftarrow \mathsf{Encapsulation}(pk)\right] \leq \delta,$$

where the expectation is over $(pk, sk) \leftarrow \mathsf{KeyGen}$ and the randomness of $\mathsf{Encapsulation}$.

Definition 4 (IND-CCA). *A KEM scheme is called* $\mathsf{IND\text{-}CCA}$ *secure if for any PPT (quantum) adversary* $\mathcal{A}$*, the advantage*

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

in the $\mathsf{Game}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM},\mathcal{A}}$ *in Fig.2 is negligible in security parameter* κ*. The oracle* $O_{\mathsf{Decaps}}(sk, \cdot \neq c^*)$ *returns* $\mathsf{Decaps}(sk, c)$ *for all* $c \neq c^*$*.*

3 New Generic NTRU Encryption paradigm

This section introduces an encoding approach through zero divisors, leading to an NTRU encryption paradigm based on encoding that unifies past encoding-based NTRU schemes from a high-level perspective. We reduce the problem of constructing a more compact NTRU encryption scheme to finding a better encoding that conforms to NTRU characteristics.

3.1 Zero Divisor Encoding

We begin with a generalized form of NTRU encryption that closely resembles the structure of prominent NTRU encryption schemes (e.g., [31,12,34,43]), with these schemes representing specific variants. For clarity, we do not consider embedding the message $\boldsymbol{m}$ in $\boldsymbol{s}$ or $\boldsymbol{e}$ during encryption, as this would necessitate additional security proofs, impose distribution constraints on $\boldsymbol{s}$ and $\boldsymbol{e}$, or potentially lead to a weaker reduction factor, such as NEV$'$ [51] under the ssp Ring-LWE assumption.

Let $\boldsymbol{h} = (\boldsymbol{g}\boldsymbol{f}^{-1}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$ be the public key. For a given message $\boldsymbol{m}$, the ciphertext is calculated as $\boldsymbol{c} \equiv (\boldsymbol{hs} + \boldsymbol{e} + p^{-1} \cdot \boldsymbol{m}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$, where p is an integer coprime to q. The decryption process proceeds as follows:

$$\boldsymbol{m}' \equiv (p \cdot \boldsymbol{cf}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q) \equiv (p \cdot (\boldsymbol{gs} + \boldsymbol{fe}) + \boldsymbol{fm}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$$

If $\lfloor (p \cdot (\boldsymbol{gs} + \boldsymbol{fe}) + \boldsymbol{fm}) \rceil_q = 0$, we can transform this equation from $\mathbb{Z}_q$ to $\mathbb{Z}$, effectively encoding the message in the lower bits of the ciphertext using the small modulus p. Converting to $\mathbb{Z}_p$ eliminates all terms except $\boldsymbol{m}$, thus recovering the message. The correct decryption condition is $\|p \cdot (\boldsymbol{gs} + \boldsymbol{fe}) + \boldsymbol{fm}\|_\infty < \frac{q}{2}$. However, the integer p amplifies the standard deviation of the noise by at least a factor of 2.

Analyzing the decryption process more carefully: First, we transform the equation from $\mathcal{R}_{\boldsymbol{\phi},q}$ to $\mathcal{R}_{\boldsymbol{\phi}}$, which is guaranteed by the correct decryption condition. Then, we convert from $\mathcal{R}_{\boldsymbol{\phi}}$ to $\mathcal{R}_{\boldsymbol{\phi},p}$, where all terms except the message become zero over $\mathcal{R}_{\boldsymbol{\phi},p}$. We can exploit a zero divisor $\boldsymbol{t}$ in a subring of $\mathcal{R}_{\boldsymbol{\phi},p}$ to extract $\boldsymbol{m} \bmod (\boldsymbol{t}, \mathbb{Z}_p)$. For example, with $\boldsymbol{\phi} = x^n + 1$, $p = 2$, and $\boldsymbol{t} = x^{n/2} + 1$, we have $\boldsymbol{\phi} = \boldsymbol{t}^2$ over $\mathbb{Z}_2$ when n is a power of 2. This enables recovery of $\boldsymbol{m} \bmod (\boldsymbol{t}, \mathbb{Z}_2)$ by encrypting as $\boldsymbol{c} = (\boldsymbol{hs} + \boldsymbol{e} + \boldsymbol{t}^{-1}\boldsymbol{m}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$. Moreover, using $\boldsymbol{t} = x^{n/2} + 1$ increases the standard deviation by only a factor of $\sqrt{2}$, improving the probability of satisfying the correct decryption condition compared to previous encoding methods.

3.2 Double Encoding Paradigm

Building on zero divisor encoding, we leverage this concept to present a general paradigm for NTRU encryption. The public parameters comprise two coprime integers $q, p \in \mathbb{N}$ and three polynomials $\boldsymbol{\phi}, \boldsymbol{t}$, and $\boldsymbol{w} \in \mathbb{Z}[x]$, such that $\boldsymbol{\phi} \equiv 0 \pmod{(\boldsymbol{t}, \mathbb{Z}_p)}$. The degrees of polynomials $\boldsymbol{\phi}$, $\boldsymbol{t}, \boldsymbol{w}$, and the message polynomial $\boldsymbol{m}$ are n, l_t, l_w, and l_m, respectively, with the constraint that $l_w + l_m \leq l_t$. Additionally, the private key $\boldsymbol{f}$ must be invertible over $\mathcal{R}_{\boldsymbol{\phi},q}$ and $\mathcal{R}_{\boldsymbol{t},p}$.

To encrypt the message polynomial $\boldsymbol{m}$, we compute the ciphertext as:

$$\boldsymbol{c} \equiv (\boldsymbol{hs} + \boldsymbol{e} + \boldsymbol{t}^{-1}\boldsymbol{wm}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$$

For decryption, we calculate $\boldsymbol{c}' \equiv (\boldsymbol{ctf}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$, which simplifies as:

$$\boldsymbol{c}' \equiv (\boldsymbol{t}(\boldsymbol{gs} + \boldsymbol{fe}) + \boldsymbol{fwm}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$$

Next, we transform this expression from $\mathcal{R}_{\boldsymbol{\phi},q}$ to $\mathcal{R}_{\boldsymbol{\phi}}$:

$$\boldsymbol{c}' \equiv (\boldsymbol{t}(\boldsymbol{gs} + \boldsymbol{fe}) + \boldsymbol{fwm} - q \cdot \boldsymbol{e}') \bmod (\boldsymbol{\phi}, \mathbb{Z})$$

where $\boldsymbol{e}' = \lfloor \boldsymbol{t}(\boldsymbol{gs} + \boldsymbol{fe}) + \boldsymbol{fwm} \rceil_q$. Subsequently, we transform it to $\mathcal{R}_{\boldsymbol{t},\mathbb{Z}_p}$:

$$\boldsymbol{c}' \equiv (\boldsymbol{fwm} - q \cdot \boldsymbol{e}') \bmod (\boldsymbol{t}, \mathbb{Z}_p)$$

Finally, we compute $\boldsymbol{m}' = (\boldsymbol{c}'\boldsymbol{f}^{-1}) \bmod (\boldsymbol{t}, \mathbb{Z}_p)$, yielding:

$$\boldsymbol{m}' \equiv (\boldsymbol{wm} - q \cdot \boldsymbol{e}'\boldsymbol{f}^{-1}) \bmod (\boldsymbol{t}, \mathbb{Z}_p)$$

The choice of $\boldsymbol{w}$ varies with l_t and l_m, leading to different decryption strategies. When $l_m = l_t$, $\boldsymbol{w}$ must be 1, requiring $\boldsymbol{e}' = 0$ for successful decryption. When $l_m < l_t$, we can utilize the redundant portion to select a $\boldsymbol{w}$ that facilitates decoding the error term $q \cdot \boldsymbol{e}'\boldsymbol{f}^{-1}$. The paradigm is summarized in Figure 3.

Fig. 3: **Generic NTRU Encryption Paradigm with Double Encoding**

```
KeyGen:
1 : $\boldsymbol{f} \leftarrow \mathbf{Sample}(\mathcal{X}_f^n)$
2 : $\boldsymbol{g} \leftarrow \mathbf{Sample}(\mathcal{X}_g^n)$
3 : if $\boldsymbol{f}$ or $\boldsymbol{g}$ is not invertible on $\mathcal{R}_{\boldsymbol{\phi},q}$
        go back to the step 1
4 : if $\boldsymbol{f}$ is not invertible on $\mathcal{R}_{\boldsymbol{t},p}$
        go back to the step 1
5 : $\boldsymbol{f}_p \leftarrow (\boldsymbol{f}^{-1}) \bmod (\boldsymbol{t}, \mathbb{Z}_p)$
6 : $\boldsymbol{h} \leftarrow (\boldsymbol{g}\boldsymbol{f}^{-1}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$

Encrypt($\boldsymbol{h}, \boldsymbol{m}$):
1 : $\boldsymbol{s} \leftarrow \mathbf{Sample}(\mathcal{X}_s^n)$
2 : $\boldsymbol{e} \leftarrow \mathbf{Sample}(\mathcal{X}_e^n)$
3 : $\boldsymbol{c} \leftarrow \left(\boldsymbol{hs} + \boldsymbol{e} + \boldsymbol{t}^{-1}\boldsymbol{wm}\right) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$

Decrypt($\boldsymbol{f}, \boldsymbol{f}_p, \boldsymbol{c}$):
1 : $\boldsymbol{c}' \leftarrow (\boldsymbol{ctf}) \bmod (\boldsymbol{\phi}, \mathbb{Z}_q)$
2 : $\boldsymbol{m}' \leftarrow (\boldsymbol{c}'\boldsymbol{f}_p) \bmod (\boldsymbol{t}, \mathbb{Z}_p)$
3 : $\boldsymbol{m} \leftarrow \mathbf{Decode}_{\boldsymbol{w}}(\boldsymbol{c}', \boldsymbol{m}', \boldsymbol{f}_p)$
```

In this paradigm, $\boldsymbol{w}$ encodes $\boldsymbol{m}$, and $\mathbf{Decode}_{\boldsymbol{w}}$ corrects errors to recover $\boldsymbol{m}$. The polynomials $\boldsymbol{t}$ and $\boldsymbol{w}$ establish a two-layer encoding structure: $\boldsymbol{t}$ eliminates the term $\boldsymbol{gs} + \boldsymbol{fe}$, while $\boldsymbol{w}$ corrects the small error term $\boldsymbol{e}'$.

The successful decryption condition is $\lfloor \boldsymbol{t}(\boldsymbol{gs} + \boldsymbol{fe}) + \boldsymbol{fwm} \rceil_q \in \mathcal{C}_{\boldsymbol{w}}$, where $\mathcal{C}_{\boldsymbol{w}}$ represents the set of errors that $\mathbf{Decode}_{\boldsymbol{w}}$ can correct. Traditional NTRU encryption schemes (e.g., [31,34,12]) can be interpreted as instantiations with $\boldsymbol{t} = 3$, $\boldsymbol{w} = 1$, and no decoding step. NEV [51] can be viewed as a variant that uses $\boldsymbol{t} = 2$ and $\boldsymbol{w} = \sum_{i=0}^{k-1} x^{in/k}$, combining high-bit encoding with k-repetition codes in its $\mathbf{Decode}_{\boldsymbol{w}}$ function to achieve the great ciphertext compactness.

3.3 Balance between Two Layers

We now adapt the generic NTRU paradigm to the power-of-two cyclotomic ring $\mathbb{Z}[x]/(x^n + 1)$ with $n = 2^l$, a prevalent instantiation in lattice cryptography, seeking to optimize $\boldsymbol{t}$ and $\boldsymbol{w}$ for maximum compactness. Within this paradigm, achieving a more compact NTRU encryption reduces to selecting optimal polynomials $\boldsymbol{t}$ and $\boldsymbol{w}$. An optimal selection minimizes both $\|\boldsymbol{t}\|$ and $\|\boldsymbol{w}\|$ while maximizing the error-correction capacity of $\boldsymbol{w}$, corresponding to selecting $\boldsymbol{w}$ with the largest possible degree l_w that the paradigm permits.

Since $\boldsymbol{t}$ must divide $\boldsymbol{\phi} = x^n + 1$ under $\mathbb{Z}_p$, and for $p = 2$ we have $x^n + 1 = (x+1)^n$, the minimal-norm divisor takes the form $x^{2^i} + 1$ for $0 < i < l$, achieving the theoretical minimum norm $\|\boldsymbol{t}\| = \sqrt{2}$. Given the constraint $l_w + l_m \leq l_t$, we maximize l_t to allow the largest possible l_w, thereby improving error tolerance or enlarging the message space. Thus, the optimal choice of $\boldsymbol{t}$ for $p = 2$ is $x^{n/2} + 1$. Furthermore, we select $\boldsymbol{w}$ such that $\boldsymbol{t} \equiv 0 \pmod{(\boldsymbol{w}, \mathbb{Z}_p)}$. Under this condition, we can directly compute $(\boldsymbol{e}') \bmod (\boldsymbol{w}, \mathbb{Z}_p)$ as $(-\frac{1}{q} \cdot \boldsymbol{c}') \bmod (\boldsymbol{w}, \mathbb{Z}_p)$, significantly simplifying and accelerating the decoding process. This optimization is only possible when $\boldsymbol{t}$ is divisible under $\mathbb{Z}_p$.

To quantify the trade-off between $\boldsymbol{t}$ and $\boldsymbol{w}$, we define:

- $\omega_1 \triangleq \|\boldsymbol{t}\|$, the primary expansion factor: a larger ω_1 necessitates more errors.
- $\omega_2 \triangleq \frac{l_w}{l_m}$, the encoding-redundancy ratio: a larger ω_2 provides greater error tolerance.

Let $\mathcal{M}$ be the message space, with $l_m = \lceil \log_p(|\mathcal{M}|) \rceil$. Table 5 presents optimal choices of $\boldsymbol{t}$ and corresponding values of ω_1 and ω_2 across various small primes p for $\mathcal{M} = \{0,1\}^{n/4}$. We omit explicit $\boldsymbol{w}$ choices to avoid diverting into efficiency comparisons among different decoding algorithms. The "Simple Decoding" column indicates whether a polynomial $\boldsymbol{w}$ dividing $\boldsymbol{t}$ over $\mathbb{Z}_p$ exists.

Table 5: Encoding Settings Comparison

$\boldsymbol{t}$	p	ω_1	ω_2	Simple Decoding (✔/✘)
$x^{n/2} + 1$	2	$\sqrt{2}$	1	✔
$x^{n/2} + x^{n/4} - 1$	3	$\sqrt{3}$	2.17	✘
$x^{n/2} + 2$	5	$\sqrt{5}$	3.64	✘
$x^{n/4} + x^{n/8} - 1$	7	$\sqrt{3}$	1.81	✘
$x^{n/2} + 3x^{n/4} + 1$	7	$\sqrt{11}$	4.61	✔

Based on our previous analysis and the data in Table 5, we recommend $\boldsymbol{t} = x^{n/2}+1$ with $p = 2$ as the optimal setting. When combined with $\boldsymbol{w} = x^{n/4}+1$, this configuration can correct a single error in $\boldsymbol{e}'$, i.e., $\mathcal{C}_{\boldsymbol{w}} = \{0\} \cup \{\pm x^i\}_{0 \leq i < n}$. For $\boldsymbol{t} = x^{n/2} + x^{n/4} - 1$ with $p = 3$, the larger ω_1 necessitates correcting two errors to match the compactness of the first choice. However, since no $\boldsymbol{w}$ divides this $\boldsymbol{t}$ over $\mathbb{Z}_3$, decoding becomes more computationally expensive, and other parameter sets offer even less favorable trade-offs.

In conclusion, we have reframed NTRU encryption design as an efficient coding-construction problem. The open question remains whether our polynomial selection represents the best choice within the double encoding paradigm.

4 DAWN Algorithms

In this section, we introduce two schemes, **DAWN.PKE** and **DAWN.KEM**, and analyze the decryption failure probability and parameter selections.

4.1 Algorithms of DAWN.PKE

Public Parameters. The structure of the large ring is fixed as $\mathcal{R}_{\boldsymbol{\phi}} = \mathcal{R}_{x^n+1}$, where n is a power-of-2 to adapt the NTT acceleration. We fix the small ring as $\mathcal{R}_{\boldsymbol{t},p} = \mathcal{R}_{x^{n/2}+1,2}$, thus $\boldsymbol{t} = x^{n/2}+1$, $p = 2$, and use $\boldsymbol{w} = x^{n/4}+1$. In summary, we have the following public parameters:

- $n \in \mathbb{N}$, the degree of $\boldsymbol{\phi}$, a power-of-2
- $q \in \mathbb{N}$, the large modulus, a prime integer
- $k_g, k_f \in \mathbb{N}$, forming the distributions of $\boldsymbol{g}$, $\boldsymbol{f}$
- $k_s, k_e \in \mathbb{N}$, forming the distributions of $\boldsymbol{s}$, $\boldsymbol{e}$
- $d_c \in \mathbb{N}$, forming the scale of compression of ciphertext
- $\mathcal{M}$, the message space $\{0,1\}^{n/4}$

Fast Inversion. In addition to performing polynomial operations on $R_{x^n+1,q}$, we also need to compute polynomial multiplication and inversion on $R_{x^{n/2}+1,2}$. While the NTT is unsuitable for the quotient ring $R_{x^{n/2}+1,2}$, multiplication in this ring is efficient due to using XOR operations. However, direct inversion over $R_{x^{n/2}+1,2}$ is computationally expensive. To address this challenge, we exploit the mathematical property that $x^{n/4}+1 = (x+1)^{n/4}$ over $\mathbb{Z}_2$ (since $n/4$ is a power-of-2), which enables a more efficient inversion process using principles similar to those in the Hensel lifting lemma. The detailed implementation of this approach is presented in Algorithm 1.

Algorithm 1 FastInversion

```
Input: a polynomial $\boldsymbol{f}$, an integer $n$, where $n$ is a power-of-2
Output: the inverse polynomial $\boldsymbol{f_2} \equiv \boldsymbol{f}^{-1} \bmod (x^n+1, \mathbb{Z}_2)$ or $\perp$ for not invertible
if $\boldsymbol{f} \equiv 0 \bmod (x+1, \mathbb{Z}_2)$ then
    return $\perp$
end if
$l \leftarrow \log_2(n)$
$\boldsymbol{k} \leftarrow (\boldsymbol{f}+1)/(x+1)$
$\boldsymbol{f_2} \leftarrow 1$
for $i$ from $0$ to $l-1$ then
    $\boldsymbol{b} \leftarrow (\boldsymbol{k}\boldsymbol{f_2}) \bmod (x^{2^i}+1, \mathbb{Z}_2)$
    $\boldsymbol{k} \leftarrow (\boldsymbol{k} + \boldsymbol{f}\boldsymbol{b}) \bmod (x^n+1, \mathbb{Z}_2)$
    $\boldsymbol{k} \leftarrow \boldsymbol{k}/(x^{2^i}+1)$
    $\boldsymbol{f_2} \leftarrow \boldsymbol{f_2} + (x^{2^i}+1)\boldsymbol{b}$
end for
return $\boldsymbol{f_2}$
```

Simple Decoding. For the encoding polynomials $\boldsymbol{t} = x^{n/2}+1$ and $\boldsymbol{w} = x^{n/4}+1$, and using $p = 2$ in DAWN.PKE and DAWN.KEM, we employ a simple decoding method that can correct a single error. We use the same notations as in Section 3.2. Since we only need to consider the case where a single error occurs, that is, when $\boldsymbol{e}' \in \{0\} \cup \{\pm x^i\}_{0 \le i < n}$, we can readily deduce that $\boldsymbol{m}' \equiv \boldsymbol{e}' \mod (x^{n/4}+1, \mathbb{Z}_2)$. However, to recover the message $\boldsymbol{m}$ correctly, we must compute $\boldsymbol{e}' \mod (x^{n/2}+1, \mathbb{Z}_2)$. If $\boldsymbol{e}' \equiv 0 \mod (x^{n/4}+1, \mathbb{Z}_2)$, then $\boldsymbol{m}$ is obtained directly. Otherwise, we express $\boldsymbol{e}'$ as $r \cdot x^i$ with $r \in \{-1, 1\}$. Under $\mathcal{R}_{x^n+1,2}$, the elements $x^i, x^{i+n/4}, x^{i+2n/4}, x^{i+3n/4}$ are indistinguishable; therefore, we substitute the error into the polynomial over $\mathcal{R}_{x^n+1}$. For the correct index i, the value $\min(|\boldsymbol{c}'_{[i]}+q|, |\boldsymbol{c}'_{[i]}-q|)$ should be the smallest among those at the indices $i+jn/4$ for $j \in \{0, 1, 2, 3\}$. The decoding fails if this minimum does not correspond to the correct index; we consider this probability in our decryption failure analysis in Section 4.3. The detailed decoding algorithm is provided in Algorithm 2.

Algorithm 2 SimpleDecoding

```
Input: three polynomials $\boldsymbol{c}', \boldsymbol{m}', \boldsymbol{f_2} \in \mathcal{R}_{x^{n/2}+1}$
Output: the message polynomial $\boldsymbol{m}$
$\boldsymbol{e}' \leftarrow (\boldsymbol{c}') \bmod (x^{n/4}+1, \mathbb{Z}_2)$
$\boldsymbol{c}'' \leftarrow ((\boldsymbol{e}')^{-1}\boldsymbol{c}') \bmod (x^n+1, \mathbb{Z})$
$i \leftarrow 0$
$base \leftarrow q$
for $j$ from 0 to 3 then
    $v \leftarrow \min(|\boldsymbol{c}''_{[jn/4]}+q|, |\boldsymbol{c}''_{[jn/4]}-q|)$
    if $v < base$ then
        $base \leftarrow v$
        $i \leftarrow jn/4$
    end if
end for
$\boldsymbol{m} \leftarrow (\boldsymbol{m}' + x^i \boldsymbol{f_2} \boldsymbol{e}') \bmod (x^{n/2}+1, \mathbb{Z}_2)$
if $\boldsymbol{e}' = 0$ then
    $\boldsymbol{m} \leftarrow (\boldsymbol{m}') \bmod (x^{n/4}, \mathbb{Z}_2)$
else
    $\boldsymbol{m} \leftarrow (\boldsymbol{m}) \bmod (x^{n/4}, \mathbb{Z}_2)$
end if
return $\boldsymbol{m}$
```

The key generation, encryption, and decryption algorithms for the PKE scheme are outlined in Algorithms 3, 4, and 5, respectively. Due to the relation $\boldsymbol{t} = \boldsymbol{w}^2$ in $\mathbb{Z}_2$, we compute the inverse of $\boldsymbol{f}$ in the ring $\mathcal{R}_{\boldsymbol{w},2}$ rather than in $\mathcal{R}_{\boldsymbol{t},2}$, thereby reducing both computation time and private key storage requirements.

4.2 Algorithms of DAWN.KEM

In this part, we use the generic FO-transformation to transform DAWN.PKE in section 4.1 into a KEM scheme, named DAWN.KEM. The key generation, encryption, and decryption algorithms for the KEM scheme are outlined in Algorithms 6, 7, and 8, respectively.

Algorithm 3 DAWN.PKE.KeyGen

```
Output: the public key $pk$, the private key $sk$
 1: $\boldsymbol{g} \leftarrow \mathbf{Sample}(\mathcal{T}^n_{n,k_g})$
 2: $\boldsymbol{f} \leftarrow \mathbf{Sample}(\mathcal{T}^n_{n,k_f})$
 3: if $\boldsymbol{f}$ or $\boldsymbol{g}$ is not invertible in $\mathcal{R}_{x^n+1,q}$ then
 4:     go back to step 1
 5: end if
 6: $\boldsymbol{h} \leftarrow (\boldsymbol{g}\boldsymbol{f}^{-1}) \bmod (x^n+1, \mathbb{Z}_q)$
 7: $\boldsymbol{f_2} \leftarrow \mathbf{FastInversion}(\boldsymbol{f}, n/4)$
 8: if $\boldsymbol{f_2} = \perp$ then
 9:     go back to step 1
10: end if
11: $(pk, sk) \leftarrow (\boldsymbol{h}, (\boldsymbol{f}, \boldsymbol{f_2}))$
12: return $(pk, sk)$
```

Algorithm 4 DAWN.PKE.Encrypt

```
Input: the public key $pk$, the message $\boldsymbol{m} \in \mathcal{M}$, the seed $\rho$
Output: the ciphertext $\boldsymbol{c}$
$\boldsymbol{h} \leftarrow pk$
$(\rho_s, \rho_e) \leftarrow \rho$
$\boldsymbol{s} \leftarrow \mathbf{Sample}(\mathcal{T}^n_{n,k_s}, \rho_s)$
$\boldsymbol{e} \leftarrow \mathbf{Sample}(\mathcal{T}^n_{n,k_e}, \rho_e)$
$\boldsymbol{c} \leftarrow \lfloor(\boldsymbol{h}\boldsymbol{s} + \boldsymbol{e} + (x^{n/4}+1)^{-1}\boldsymbol{m}) \bmod (x^n+1, \mathbb{Z}_q)\rceil_{d_c}$
return $\boldsymbol{c}$
```

Algorithm 5 DAWN.PKE.Decrypt

```
Input: the ciphertext $\boldsymbol{c}$, the private key $sk$
Output: the message $\boldsymbol{m}$
$(\boldsymbol{f}, \boldsymbol{f_2}) \leftarrow sk$
$\boldsymbol{c}' \leftarrow (d_c \cdot (x^{n/2}+1)\boldsymbol{c}\boldsymbol{f}) \bmod (x^n+1, \mathbb{Z}_q)$
$\boldsymbol{m}' \leftarrow (\boldsymbol{c}'\boldsymbol{f_2}) \bmod (x^{n/2}+1, \mathbb{Z}_2)$
$\boldsymbol{m} \leftarrow \mathbf{SimpleDecoding}(\boldsymbol{c}', \boldsymbol{m}', \boldsymbol{f_2})$
return $\boldsymbol{m}$
```

Algorithm 6 DAWN.KEM.KeyGen

```
Output: the public key $pk$, the private key $sk$
$(pk, sk') \leftarrow$ DAWN.PKE.KeyGen
$\mathrm{K}' \leftarrow \mathbf{Sample}(\mathcal{U}(\mathcal{M}))$
$\mathrm{H}_{pk} \leftarrow \mathbf{Hash}_{pk}(pk)$
$sk \leftarrow (\mathrm{K}', sk', pk, \mathrm{H}_{pk})$
return $(pk, sk)$
```

Algorithm 7 DAWN.KEM.Encapsulation

```
Input: the public key $pk$
Output: the ciphertext $\boldsymbol{c}$, the shared key $\mathrm{K}$
$\boldsymbol{m} \leftarrow \mathcal{M}$
$(\overline{\mathrm{K}}, \rho) \leftarrow \mathbf{Hash}_m(m, \mathbf{Hash}_{pk}(pk))$
$\boldsymbol{c} \leftarrow \mathbf{DAWN.PKE.Encrypt}(pk, \boldsymbol{m}, \rho)$
$\mathrm{K} \leftarrow \mathbf{Hash}_{\mathrm{K}}(\overline{\mathrm{K}}, \boldsymbol{c})$
return $(\boldsymbol{c}, \mathrm{K})$
```

Algorithm 8 DAWN.KEM.Decapsulation

```
Input: the ciphertext $\boldsymbol{c}$, the private key $sk$
Output: the shared key $\mathrm{K}$
$(\mathrm{K}', sk', pk, \mathrm{H}_{pk}) \leftarrow sk$
$\boldsymbol{m} \leftarrow \mathbf{DAWN.PKE.Decrypt}(\boldsymbol{c}, sk')$
$(\overline{\mathrm{K}}, \rho) \leftarrow \mathbf{Hash}_m(\boldsymbol{m}, \mathrm{H}_{pk})$
if $\boldsymbol{c} = \mathbf{DAWN.PKE.Encrypt}(pk, \boldsymbol{m}, \rho)$ then
    $\mathrm{K} \leftarrow \mathbf{Hash}_{\mathrm{K}}(\overline{\mathrm{K}}, \boldsymbol{c})$
else
    $\mathrm{K} \leftarrow \mathbf{Hash}_{\mathrm{K}}(\mathrm{K}', \boldsymbol{c})$
end if
return K
```

4.3 Analysis of Decryption Failure

We classify decryption failures into two events:

- $\mathbb{E}_1$: the error $\boldsymbol{e}'$ lies outside the valid set, i.e. $\boldsymbol{e}' \notin \{0\} \cup \{\pm x^i\}_{0 \le i < n}$.
- $\mathbb{E}_2$: the **SimpleDecoding** algorithm selects an incorrect index.

Let $\boldsymbol{z} \triangleq (x^{n/2}+1)\big(\boldsymbol{g}\boldsymbol{s}+\boldsymbol{f}(\boldsymbol{e}+\boldsymbol{e}_{d_c})\big)+(x^{n/4}+1)\boldsymbol{f}\boldsymbol{m},$, where $\boldsymbol{e}_{d_c}$ is the rounding error from d_c-compression, such that $\lfloor \boldsymbol{z} \rceil_q = \boldsymbol{e}'$. We assume that the coefficients of $\boldsymbol{z}$ are independent and follow a similar calculation approach to that used for Kyber's decryption failure estimation script[2]. We employ the script from the repository[3] to determine the distribution of the coefficients of $\boldsymbol{z}$, and then estimate the probability of $\mathbb{E}_1$ as follows:

1. Compute $\mathcal{D}_z$, the distribution of the coefficients of $\boldsymbol{z}$.
2. Let δ_1 be the probability that an element drawn from $\mathcal{D}_z$ has absolute value exceeds $\frac{q-1}{2}$.
3. Calculate $\mathcal{P}_1 = \sum_{i=2}^{n} \binom{n}{i} \delta_1^i (1-\delta_1)^{n-i}$, the probability of $\mathbb{E}_1$

For the failed decoding event $\mathbb{E}_2$, we consider the conservative condition that $|\boldsymbol{c}'_{[i]}| > \min\big(|\boldsymbol{c}'_{[i+jn/4]} + q|, |\boldsymbol{c}'_{[i+jn/4]} - q|\big)$, $j \in \{1, 2, 3\}$, and estimate its probability similarly:

1. Compute $\mathcal{D}'_z$, the distribution of the sum of two independent draws from $\mathcal{D}_z$.
2. Let δ_2 be the probability that an element drawn from $\mathcal{D}'_z$ has absolute value exceeds q.
3. Calculate $\mathcal{P}_2 = \sum_{i=1}^{3n} \binom{3n}{i} \delta_2^i (1-\delta_2)^{3n-i}$, the probability of $\mathbb{E}_2$

Finally, the overall decryption failure probability satisfies $\mathcal{P} \leq \mathcal{P}_1 + \mathcal{P}_2$, and we take $\mathcal{P}_1 + \mathcal{P}_2$ as our estimate. Concrete values are given in Table 6.

4.4 Parameter Selection

Table 6 presents the recommended parameters for DAWN at the NIST-I and NIST-V security levels. We propose two parameter sets tailored to different objectives: DAWN-α minimizes ciphertext length, while DAWN-β minimizes public key length. For scenarios involving a single key generation followed by multiple encryptions, DAWN-α is preferable. Conversely, for scenarios where both the public key and ciphertext sizes are equally important, DAWN-β is recommended. We note that DAWN-α-512′ uses the same parameters as DAWN-α-512 except for $d_c = 3$, and achieves the same security level.

Table 6: Suggested parameters for DAWN

	n	q	(k_g, k_f)	(k_s, k_e)	d_c	Public Key	Cipher Text	Dec. Failure	Sec. Level
DAWN-α-512	512	769	$(160, 64)$	$(96, 160)$	7	615	436	2^{-133}	140
DAWN-α-1024	1024	769	$(256, 96)$	$(192, 256)$	4	1229	973	2^{-163}	270
DAWN-β-512	512	257	$(64, 32)$	$(48, 64)$	2	514	450	2^{-130}	134
DAWN-β-1024	1024	257	$(96, 64)$	$(96, 96)$	1	1027	1027	2^{-138}	267

[2] https://github.com/pq-crystals/security-estimates/blob/master/Kyber_failure.py

[3] https://github.com/Icarid-Liu/lattice-KEM-DFR-estimator

5 Security

5.1 Hardness Assumption

Decisional NTRU Assumption. Let $\mathcal{R}_{\boldsymbol{\phi},q}^{\times}$ be the multiplicative group of $\mathcal{R}_{\boldsymbol{\phi},q}$, and $\mathcal{X}_f, \mathcal{X}_g$ be two distributions over $\mathbb{Z}_q$. Let $\boldsymbol{f}$ and $\boldsymbol{g}$ be two polynomials drawn from $\mathcal{X}_f^n \cap \mathcal{R}_{\boldsymbol{\phi},q}^{\times}$ and $\mathcal{X}_g^n \cap \mathcal{R}_{\boldsymbol{\phi},q}^{\times}$, respectively, where $\boldsymbol{f}$ is also invertible over $\mathcal{R}_{\boldsymbol{\phi},p}$. Let $\boldsymbol{u}$ be a polynomial drawn from $\mathcal{U}(\mathcal{R}_{\boldsymbol{\phi},q}^{\times})$. The advantage of an adversary $\mathcal{A}$ in solving the decisional NTRU problem is defined as

$$\mathrm{Adv}^{\mathrm{NTRU}}_{\mathcal{R}_{\boldsymbol{\phi},q},p,\mathcal{X}_f,\mathcal{X}_g}(\mathcal{A}) = \left|\Pr[\mathcal{A}(\boldsymbol{g}\boldsymbol{f}^{\boldsymbol{-1}}) = 1] - \Pr[\mathcal{A}(\boldsymbol{u}) = 1]\right|$$

In the setting we consider, where $\boldsymbol{\phi} = x^n + 1$ with $n = 2^l$ for some $l \in \mathbb{N}$, and $p = 2, \mathcal{X}_g = \mathcal{T}_{n,k_g}, \mathcal{X}_f = \mathcal{T}_{n,k_f}$, extensive research has been devoted to the hardness of this decisional NTRU problem. In a power-of-2 cyclotomic ring, the work in [49] demonstrated that if $\mathcal{X}_g, \mathcal{X}_g$ are taken as discrete Gaussian distributions with standard deviations $\omega(n\sqrt{q})$, the distinguishing advantage is exponentially small. Moreover, [46] shows that a variant of the search NTRU problem, where $|\boldsymbol{g}|$ and $|\boldsymbol{f}|$ are constrained to be approximately $\sqrt{q}$, is at least as hard as the worst-case approximate Shortest Vector Problem on ideal lattices, thereby providing a robust security foundation. This decisional NTRU problem with a narrow distribution is also known as the decisional small polynomial ratio (DSPR) assumption and has been widely used in [40,22,27,15,28,51].

Decisional Ring-LWE Assumption. Let $\mathcal{R}_{\boldsymbol{\phi},q} = \mathbb{Z}_q[x]/\boldsymbol{\phi}$, and let $\mathcal{X}_s, \mathcal{X}_e$ be distributions over $\mathbb{Z}_q$. Sample $\boldsymbol{a} \leftarrow \mathcal{U}(\mathcal{R}_{\boldsymbol{\phi},q})$, $\boldsymbol{s} \leftarrow \mathcal{X}_s^n \cap \mathcal{R}_{\boldsymbol{\phi},q}$, $\boldsymbol{e} \leftarrow \mathcal{X}_e^n \cap \mathcal{R}_{\boldsymbol{\phi},q}$, and let $\boldsymbol{u} \leftarrow \mathcal{U}(\mathcal{R}_{\boldsymbol{\phi},q})$. The advantage of an adversary in distinguishing Ring-LWE samples from uniform is

$$\mathrm{Adv}^{\mathrm{RLWE}}_{\mathcal{R}_{\boldsymbol{\phi},q},\mathcal{X}_s,\mathcal{X}_e}(\mathcal{A}) = \left|\Pr[\mathcal{A}(\boldsymbol{a},\, \boldsymbol{a}\boldsymbol{s} + \boldsymbol{e}) = 1] - \Pr[\mathcal{A}(\boldsymbol{a},\, \boldsymbol{u}) = 1]\right|$$

In our setting, $\boldsymbol{\phi} = x^n + 1$ with $n = 2^l$, and $\mathcal{X}_s = \mathcal{T}_{n,k_s}$, $\mathcal{X}_e = \mathcal{T}_{n,k_e}$. Lyubashevsky, Peikert, and Regev introduced the Ring-LWE problem in [42], and they proved that, in power-of-two cyclotomic rings, the distinguishing advantage is negligible for suitably chosen discrete Gaussian errors. Ring LWE has been widely used in [50,16,18,7,41,51].

5.2 PKE Security

Theorem 1. *Let $\mathcal{A}$ be an* IND-CPA *adversary against the* DAWN.PKE *scheme described in Algorithms 3, 4, and 5. Then there exists an adversary $\mathcal{B}$ against the decisional NTRU assumption and an adversary $\mathcal{C}$ against the decisional Ring-LWE assumption, both with roughly the same running time as $\mathcal{A}$, such that:*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathrm{DAWN.PKE}}(\mathcal{A}) \leq \mathrm{Adv}^{\mathrm{NTRU}}_{\mathcal{R}_{\boldsymbol{\phi},q},p,\mathcal{X}_f,\mathcal{X}_g}(\mathcal{B}) + \mathrm{Adv}^{\mathrm{RLWE}}_{\mathcal{R}_{\boldsymbol{\phi},q},\mathcal{X}_s,\mathcal{X}_e}(\mathcal{C}).$$

Proof. The adversary $\mathcal{A}$ can obtain the public key pk and make encryption queries for the $\mathsf{DAWN.PKE}$ scheme. Subsequently, $\mathcal{A}$ selects two messages $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$. The challenger randomly chooses one of them and encrypts it. The adversary's goal is to determine whether the encrypted message is $\boldsymbol{m}_0$ or $\boldsymbol{m}_1$.

To prove Theorem 1, we introduce a sequence of games Game_0 through Game_2, where Game_0 represents the standard $\mathsf{IND\text{-}CPA}$ security game, and Game_2 is a random one. The security of our scheme is established by showing that Game_0 and Game_2 are computationally indistinguishable from the adversary $\mathcal{A}$'s perspective. Let $\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Game}_i}(\mathcal{A})$ denote the advantage of $\mathcal{A}$ in Game_i, where $i \in \{0, 1, 2\}$.

Game$_0$: This is the standard $\mathsf{IND\text{-}CPA}$ security game for the $\mathsf{DAWN.PKE}$ scheme described in Algorithms 3, 4, and 5.

- **Public key**: The public key $\boldsymbol{h}$ is generated according to Algorithm 3, where $\boldsymbol{h} = \boldsymbol{g}\boldsymbol{f}^{-1} \in \mathcal{R}^{\times}_{\phi,q}$.
- **Encryption query**: For any message m submitted by adversary $\mathcal{A}$, Game_0 encrypts it following Algorithm 4.
- **Challenge**: For the two messages $\boldsymbol{m}_0, \boldsymbol{m}_1$ submitted by adversary $\mathcal{A}$, Game_0 randomly selects $b \in \{0, 1\}$. It then computes the ciphertext $\boldsymbol{c}_b = \mathsf{Encrypt}(pk, \boldsymbol{m}_b)$ and sends $\boldsymbol{c}_b$ to $\mathcal{A}$. The adversary $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ and wins if $b = b'$.

By definition, we have $\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Game}_0}(\mathcal{A}) = \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{DAWN.PKE}}(\mathcal{A})$.

Game$_1$: Game_1 is identical to Game_0 except for the generation of the public key $\boldsymbol{h}$.

- **Public key**: In this game, $\boldsymbol{h}$ is sampled uniformly at random from $\mathcal{U}(\mathcal{R}^{\times}_{\phi,q})$ instead of being computed as $\boldsymbol{h} = \boldsymbol{g}\boldsymbol{f}^{-1}$.

The only difference between Game_0 and Game_1 is that in Game_0, the public key $\boldsymbol{h}$ is an NTRU instance $\boldsymbol{h} = \boldsymbol{g}\boldsymbol{f}^{-1}$, whereas in Game_1, $\boldsymbol{h}$ is uniformly random. Therefore, any advantage that $\mathcal{A}$ gains in distinguishing between these games is bounded by the decisional NTRU assumption. Thus, we have:

$$|\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Game}_0}(\mathcal{A}) - \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Game}_1}(\mathcal{A})| \leq \mathrm{Adv}^{\mathrm{NTRU}}_{\mathcal{R}_{\boldsymbol{\phi},q},p,\mathcal{X}_f,\mathcal{X}_g}(\mathcal{B})$$

Game$_2$: Game_2 is similar to Game_1 except for how the challenge ciphertext is generated.

- **Challenge**: For the challenge messages $\boldsymbol{m}_0, \boldsymbol{m}_1$ submitted by adversary $\mathcal{A}$, Game_2 randomly selects $b \in \{0, 1\}$. It then samples $\boldsymbol{u}$ uniformly at random from $\mathcal{U}(\mathcal{R}^{\times}_{\phi,q})$ and computes

$$\boldsymbol{c}_b = \left\lfloor \left(\boldsymbol{u} + (x^{n/4} + 1)^{-1}\boldsymbol{m}_b\right) \bmod (x^n + 1, \mathbb{Z}_q)\right\rceil_{d_c}$$

  instead of using $\boldsymbol{hs} + \boldsymbol{e}$ as the masking term.

The difference between Game_1 and Game_2 is that in Game_1, the masking term $\boldsymbol{hs} + \boldsymbol{e}$ is a Ring-LWE instance, whereas in Game_2, the term $\boldsymbol{u}$ is uniformly random. Any advantage that $\mathcal{A}$ gains in distinguishing between these games is bounded by the decisional Ring-LWE assumption. Thus, we have:

$$|\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Game}_1}(\mathcal{A}) - \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Game}_2}(\mathcal{A})| \leq \mathrm{Adv}^{\mathrm{RLWE}}_{\mathcal{R}_{\phi,q},\mathcal{X}_s,\mathcal{X}_e}(\mathcal{C})$$

Finally, in Game_2, since $\boldsymbol{u}$ is sampled uniformly at random from $\mathcal{U}(\mathcal{R}^{\times}_{\phi,q})$, the term $\boldsymbol{u}+(x^{n/4}+1)^{-1}\boldsymbol{m}_b$ is also uniformly distributed over $\mathcal{R}^{\times}_{\phi,q}$, regardless of the value of b. Consequently, the distributions of $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$ are computationally indistinguishable to adversary $\mathcal{A}$. This implies that $\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Game}_2}(\mathcal{A}) \leq \mathsf{negl}(\kappa)$, where κ is the security parameter.

Combining these results, we have:

$$\begin{aligned}
\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{DAWN.PKE}}(\mathcal{A}) &= \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Game}_0}(\mathcal{A}) \\
&\leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Game}_1}(\mathcal{A}) + \mathrm{Adv}^{\mathrm{NTRU}}_{\mathcal{R}_{\phi,q},p,\mathcal{X}_f,\mathcal{X}_g}(\mathcal{B}) \\
&\leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Game}_2}(\mathcal{A}) + \mathrm{Adv}^{\mathrm{RLWE}}_{\mathcal{R}_{\phi,q},\mathcal{X}_s,\mathcal{X}_e}(\mathcal{C}) + \mathrm{Adv}^{\mathrm{NTRU}}_{\mathcal{R}_{\phi,q},p,\mathcal{X}_f,\mathcal{X}_g}(\mathcal{B}) \\
&\leq \mathsf{negl}(\kappa) + \mathrm{Adv}^{\mathrm{RLWE}}_{\mathcal{R}_{\phi,q},\mathcal{X}_s,\mathcal{X}_e}(\mathcal{C}) + \mathrm{Adv}^{\mathrm{NTRU}}_{\mathcal{R}_{\phi,q},p,\mathcal{X}_f,\mathcal{X}_g}(\mathcal{B}) \\
&\leq \mathrm{Adv}^{\mathrm{NTRU}}_{\mathcal{R}_{\phi,q},p,\mathcal{X}_f,\mathcal{X}_g}(\mathcal{B}) + \mathrm{Adv}^{\mathrm{RLWE}}_{\mathcal{R}_{\phi,q},\mathcal{X}_s,\mathcal{X}_e}(\mathcal{C})
\end{aligned}$$

This completes the proof of Theorem 1.

5.3 KEM Security

The transformation from DAWN.PKE to DAWN.KEM follows the same Fujisaki–Okamoto (FO) transformation (with implicit rejection) used in Kyber [9] and NEV [51]. The IND-CCA security of DAWN.KEM in the quantum random oracle model (QROM) is therefore guaranteed [33,35], provided that DAWN.PKE is IND-CPA secure.

Theorem 2. DAWN.PKE *is a PKE scheme that is δ-correct, with message space $\mathcal{M}$. Let $\mathcal{A}$ is the* IND-CCA *adversary against* DAWN.KEM *scheme described in Algorithms 6, 7, 8, making q quantum queries to random oracles* $\mathbf{Hash}_m$, $\mathbf{Hash}_{pk}$, *and* $\mathbf{Hash}_K$, *and q_D classical queries to the decapsulation oracle* Decaps, *then there exists an* IND-CPA *adversary $\mathcal{B}$ against* DAWN.PKE *described in Algorithms 3, 4, 5 of roughly the same running time as that of $\mathcal{A}$ such that:*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathrm{DAWN.KEM}}(\mathcal{A}) \leq 2q\sqrt{\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathrm{DAWN.PKE}}(\mathcal{B})} + \frac{2q}{\sqrt{|\mathcal{M}|}} + 4q\sqrt{\delta}.$$

5.4 Concrete Security

A KEM necessitates rigorous security analysis for both the private key and the shared key. The security of the private key is fundamentally anchored in the decisional NTRU assumption, while the security of the shared key derives from the decisional Ring-LWE assumption. This section presents methodologies for analyzing the security of both Ring-LWE and NTRU, followed by concrete security evaluations utilizing state-of-the-art estimation techniques. We define **Vec** as the function that maps a polynomial $\boldsymbol{a} = \sum_{i=0}^{n-1} a_i x^i$ to a row vector $(a_i)_{0 \le i < n}$, and **Mat** as the function that maps a polynomial $\boldsymbol{a} \in \mathcal{R}_{\boldsymbol{\phi}}$ to a matrix $(\mathbf{Vec}(x^i \boldsymbol{a}))_{0 \le i < n}$. We denote matrices consisting entirely of zeros as $\mathbf{0}$.

5.4.1 Concrete Security of RLWE

We define a Ring-LWE instance as $(\boldsymbol{a}, \boldsymbol{b} = \boldsymbol{a}\boldsymbol{s} + \boldsymbol{e})$, where $\boldsymbol{a} \leftarrow \mathcal{U}(\mathcal{R}_{\boldsymbol{\phi},q})$, $\boldsymbol{s} \leftarrow \mathcal{X}_s^n$, and $\boldsymbol{e} \leftarrow \mathcal{X}_e^n$. The standard approach to solving Ring-LWE involves transforming it into LWE and applying LWE attack methodologies, except for certain weak instances identified in [26] and [17]. Extensive research has categorized LWE attack methods into three primary approaches: lattice-based, combinatorial, and algebraic techniques. Modern cryptographic schemes evaluate security by examining all these approaches and their hybrid variants.

Lattice Attacks. Lattice-based approaches constitute the most effective methodology for addressing hard problems in lattice cryptography. These attacks follow two principal strategies: primal and dual. For clarity of presentation, we omit scale factors in the following lattice bases.

Primal Attack. The primal attack constructs a lattice through embedding techniques [36][10]:

$$\mathcal{L}_{\boldsymbol{a},\boldsymbol{b},q} = \mathbb{Z}^{(2n+1)\times(2n+1)} \cdot \begin{pmatrix} \mathbf{Mat}(\boldsymbol{a}) & \mathbf{I}_n & \mathbf{0} \\ q \cdot \mathbf{I}_n & \mathbf{0} & \mathbf{0} \\ \mathbf{Vec}(\boldsymbol{b}) & \mathbf{0} & 1 \end{pmatrix}$$

In practical implementations where the standard deviations of $\mathcal{X}_s$ and $\mathcal{X}_e$ are sufficiently constrained, the vector $(\mathbf{Vec}(\boldsymbol{e}), -\mathbf{Vec}(\boldsymbol{s}), 1)$ constitutes the shortest vector in $\mathcal{L}_{\boldsymbol{a},\mathbf{b},q}$ with high probability. The challenge becomes finding this shortest vector, a task directly addressed by lattice reduction algorithms. The Block-Korkine-Zolotarev (BKZ) algorithm [47] represents the most efficient approach, with computational costs determined by the lattice basis matrix determinant, lattice dimension, and the norm of the target vector. Quantum computing enhances BKZ performance, resulting in different time complexities for classical versus quantum computational models. The algebraic structure of the ring enables accelerated lattice reduction as described in [38], providing polynomial efficiency improvements. Our analysis employs the contemporary cost model in [44].

Dual Attack. The dual attack first partitions $\mathbf{Mat}(\boldsymbol{a})$ into $\mathbf{A_0} \in \mathbb{Z}_q^{n_0 \times n}$ and $\mathbf{A_1} \in \mathbb{Z}_q^{n_1 \times n}$, while dividing $\mathbf{Vec}(\boldsymbol{s})$ into $\boldsymbol{s_0} \in \mathbb{Z}^{n_0}$ and $\boldsymbol{s_1} \in \mathbb{Z}^{n_1}$. It then constructs the lattice:

$$\mathcal{L}_{\mathbf{A_0},q} = \mathbb{Z}^{(n+n_0)\times(n+n_0)} \cdot \begin{pmatrix} \mathbf{A_0}^{\intercal} & \mathbf{I}_n \\ q\cdot\mathbf{I}_{n_0} & \mathbf{0} \end{pmatrix}$$

Applying BKZ to this lattice produces multiple short vectors $\boldsymbol{v_i}$ such that $\boldsymbol{v_i}\mathbf{A_0}^{\intercal} \equiv \boldsymbol{w_i} \pmod q$, with both $\|\boldsymbol{v_i}\|$ and $\|\boldsymbol{w_i}\|$ remaining small. This leads to the relationship $\langle \boldsymbol{v_i}, \mathbf{Vec}(\boldsymbol{b})\rangle \equiv \langle \boldsymbol{v_i}\mathbf{A_1}^{\intercal}, \boldsymbol{s_1}\rangle + \langle \boldsymbol{w_i}, \boldsymbol{s_0}\rangle + \langle \boldsymbol{v_i}, \mathbf{Vec}(\boldsymbol{e})\rangle \pmod q$. Since the term $\langle \boldsymbol{w_i}, \boldsymbol{s_0}\rangle + \langle \boldsymbol{v_i}, \mathbf{Vec}(\boldsymbol{e})\rangle$ has small magnitude, we can identify the correct $\boldsymbol{s_1}$ through enumeration. The computational cost for this attack, encompassing both lattice reduction and enumeration phases, can be calculated using methods from [29] and [44].

Combinatorial Attack. The Blum-Kalai-Wasserman (BKW) algorithm [13] represents the principal combinatorial attack methodology. While structurally similar to the dual attack, BKW substitutes lattice reduction with combinatorial computation techniques. This approach performs particularly well with specific parameter configurations. Its complexity can be assessed using the methodology outlined in [3].

Algebraic Attack. Algebraic attacks, initially proposed in [8] and further developed in [2], derive equations for variables $\boldsymbol{s}_{[i]}$ by exploiting the limited size of the set $\boldsymbol{e}_{[i]}$. This process generates n high-degree equations with n variables, which are resolved through Gröbner basis computation algorithms. The computational complexity approximates the time requirements for Gröbner basis calculation. However, algebraic approaches typically offer limited practical advantages in most implementations.

5.4.2 Concrete Security of NTRU

We define an NTRU instance as $\boldsymbol{h} = \boldsymbol{g}\boldsymbol{f}^{-1}$, where $\boldsymbol{g} \leftarrow \mathcal{X}_g^n \cap \mathcal{R}_{\boldsymbol{\phi},q}^{\times}$, and $\boldsymbol{f} \leftarrow \mathcal{X}_f^n \cap \mathcal{R}_{\boldsymbol{\phi},q}^{\times}$. Similar to Ring-LWE, the ring structure inherent in NTRU can be expanded into matrix form, where each polynomial corresponds to a matrix in $\mathbb{Z}^n$:

$$\mathcal{L}_{\boldsymbol{h},q} = \mathbb{Z}^{2n\times 2n} \cdot \begin{pmatrix} \mathbf{Mat}(\boldsymbol{h}) & \mathbf{I}_n \\ q\cdot\mathbf{I}_n & \mathbf{0} \end{pmatrix} = \mathbb{Z}^{2n\times 2n} \cdot \begin{pmatrix} \mathbf{Mat}(\boldsymbol{g}) & \mathbf{Mat}(\boldsymbol{f}) \\ \mathbf{Mat}(\boldsymbol{G}) & \mathbf{Mat}(\boldsymbol{F}) \end{pmatrix}$$

Distinguishing $\boldsymbol{h}$ from $\boldsymbol{u} \leftarrow \mathcal{U}(\mathcal{R}_{\boldsymbol{\phi},q})$ requires distinguishing the lattice $\mathcal{L}_{\boldsymbol{h},q}$ and the lattice $\mathcal{L}_{\boldsymbol{u},q}$, specifically whether they contain short bases. Unlike Ring-LWE scenarios where the lattice contains a single short vector, NTRU lattices feature n short vectors of the form $(x^i\boldsymbol{g}, x^i\boldsymbol{f})$. Research in [23] proposed a dense lattice attack exploiting this property for large moduli (approximately $\mathcal{O}(n^{2.484})$). However, our scheme employs smaller moduli, rendering this attack ineffective. We can reformulate NTRU as $\boldsymbol{h}\boldsymbol{f} - \boldsymbol{g} = 0$, thus viewing it as a special case of Ring-LWE where $\boldsymbol{b} = 0$. Consequently, all attacks applicable to Ring-LWE can be adapted to NTRU, except for the dual attack [4].

5.4.3 Concrete Security of DAWN

Building on our analysis, we employ the lattice estimator[4] (up to commit 5ba00f5) of [5] to derive concrete security parameters for DAWN. This tool provides strength estimates for both Ring-LWE and NTRU instantiations, as summarized in Table 7. For each attack strategy, we compute its computational cost and adopt the minimum value as our concrete security level, following established practice in lattice-based cryptography.

Table 7: Concrete security of DAWN (in bits)

Scheme	Cost Model	Private Key (NTRU)	Shared Key (RLWE)
DAWN-α-512	classical	139.5	140.4
	quantum	133.5	134.8
DAWN-α-1024	classical	269.9	270.5
	quantum	250.8	254.2
DAWN-β-512	classical	134.3	136.9
	quantum	134.1	135.4
DAWN-β-1024	classical	266.5	268.7
	quantum	251.2	252.5

6 Implementation

6.1 Number Theoretic Transform

The Number Theoretic Transform (NTT) is a specialized adaptation of the Fast Fourier Transform for finite fields, providing an elegant and efficient approach to polynomial operations in lattice-based cryptography. For polynomial multiplication and inversion in $\mathcal{R}_q$, the NTT reduces computational complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$, representing a significant improvement in algorithmic efficiency.

Consider a polynomial ring $\mathcal{R}_{x^n+1,q}$, where $n = 2^l$ for $l \in \mathbb{N}$, and let ζ denote a $2n$-th primitive root of unity in $\mathbb{Z}_q$. The foundation of the NTT is the following ring isomorphism:

$$\mathbb{Z}_q[x]/(x^n + 1) \cong \mathbb{Z}_q[x]/(x^{n/2} - \zeta^{n/2}) \times \mathbb{Z}_q[x]/(x^{n/2} + \zeta^{n/2})$$

This isomorphism allows for recursive decomposition, as both factors $(x^{n/2} - \zeta^{n/2})$ and $(x^{n/2} + \zeta^{n/2})$ can be further decomposed until reaching n linear polynomials. When $\mathbb{Z}_q$ does not contain a $2n$-th primitive root of unity, we employ a $\frac{2n}{d}$-th primitive root instead, which yields $\frac{n}{d}$ polynomials of degree d, where d divides n. This distinguishes between two variants of the algorithm:

[4] https://github.com/malb/lattice-estimator

- *Complete NTT*: When $d = 1$, allowing full decomposition into linear factors
- *Incomplete NTT*: When $d > 1$, resulting in partial decomposition

In our DAWN cryptosystem implementations, DAWN-α-512 (resp. DAWN-β-512) uses $d = 4$, while DAWN-α-1024 (resp. DAWN-β-1024) employs $d = 8$. After decomposition, polynomial operations are performed on individual factors using standard techniques, followed by application of the Inverse Number Theoretic Transform (INTT) to reconstruct the result in the original ring. Furthermore, we incorporate the lazy reduction technique described by Seiler [48] to enhance computational efficiency.

Our implementation leverages vectorization through Single Instruction Multiple Data (`SIMD`) architectures. Beyond a standard `C` implementation, we developed an optimized version using the `AVX2` instruction set, which supports 256-bit wide `SIMD` registers. This architecture enables parallel processing of multiple coefficients within a single instruction. Given that our selected parameters require a 16-bit word size for coefficient storage, the `AVX2` implementation can process 16 coefficients simultaneously in each vector register, substantially accelerating computation throughput.

During key generation, we must compute the inverse of $\boldsymbol{f}$ over $\mathcal{R}_{x^n+1,q}$. However, the NTT incompleteness is prohibitively high in DAWN-α-1024 and DAWN-β-1024, where the bottom polynomials have degree 8. Direct polynomial inversion under such conditions would be computationally expensive. We implement the splitting technique proposed in BAT [28] to address this challenge.

For a polynomial $\boldsymbol{f} \in \mathcal{R}_{x^n+1,q}$, we define $\boldsymbol{f_0} \triangleq \sum_{i=0}^{n/2-1} \boldsymbol{f}_{[2i]} x^i$ and $\boldsymbol{f_1} \triangleq \sum_{i=0}^{n/2-1} \boldsymbol{f}_{[2i+1]} x^i$. Using even-odd splitting, we can rewrite $\boldsymbol{f}$ as $\boldsymbol{f}(x) = \boldsymbol{f_0}(x^2) + x\boldsymbol{f_1}(x^2)$. Consequently, the inverse can be expressed as:

$$\frac{1}{\boldsymbol{f}} = \frac{\boldsymbol{f_0}(x^2) - x\boldsymbol{f_1}(x^2)}{({\boldsymbol{f_0}}^2 - x^2{\boldsymbol{f_1}}^2)(x^2)}$$

This transformation converts the challenge of computing the inverse of $\boldsymbol{f}$ over $\mathcal{R}_{x^n+1,q}$ into computing the inverse of ${\boldsymbol{f_0}}^2 - x^2{\boldsymbol{f_1}}^2$ over $\mathcal{R}_{x^{n/2}+1,q}$, followed by a single polynomial multiplication. Through this approach, we reduce the inversion calculation from $\mathcal{R}_{x^{1024}+1,q}$ to $\mathcal{R}_{x^{128}+1,q}$, where complete NTT can be efficiently employed. We note that this acceleration provides no computational advantage for cases where $d = 4$.

6.2 Storage

To minimize storage requirements, we implement a systematic compression methodology for all component polynomials. Public keys and ciphertexts consist of polynomials with coefficients constrained to specific ranges. By treating these polynomials as vectors and strategically combining multiple coefficients, we optimize byte utilization across the system.

We first present the general form of our coefficient combination approach. For a polynomial $\boldsymbol{f} \in \mathcal{R}_{x^n+1}$, each coefficient is bounded within the range $[-r, r]$.

We select a modulus $N \geq 2r+1$ for the combination, ensuring that each element from $[-r, r]$ maps injectively to $\mathbb{Z}_N$. Writing N in its unique prime factorization form as $N = p_1^{i_1} \cdots p_t^{i_t}$, we transform $\boldsymbol{f}$ into $\mathbb{Z}_{p_1}^{i_1} \times \cdots \times \mathbb{Z}_{p_t}^{i_t}$ using CRT.

For each prime factor p_j, we determine an integer u_j that maximizes $\log_2(p_j^{u_j}) - \lfloor \log_2(p_j^{u_j}) \rfloor$. This optimization allows us to combine each set of u_j elements over $\mathbb{Z}_{p_j}$ into a single element requiring $\lceil \log_2(p_j^{u_j}) \rceil$ bits. For the remaining ($i_j n \bmod u_j$) elements over $\mathbb{Z}_{p_j}$, we combine them into a single element of $\lceil \log_2(p_j^{(i_j n \bmod u_j)}) \rceil$ bits. While this methodology theoretically maximizes storage utilization efficiency, practical considerations necessitate certain constraints. In ideal conditions, we would select large values for u_j to maximize $\log_2(p_j^{u_j}) - \lfloor \log_2(p_j^{u_j}) \rfloor$, such as when $p_j = 257$. However, to ensure efficient execution with limited bit length, we impose the constraint $\lceil \log_2(p_j^{u_j}) \rceil \leq 64$, enabling the use of standard 64-bit multiplications. Furthermore, we strategically combine certain prime factors p_j to enhance computational practicality, as this approach reduces the total number of coefficient combinations required during processing.

The DAWN family comprises two parameter sets: DAWN-512 and DAWN-1024. Due to rounding operations in the encapsulation process, each variant involves two distinct polynomial types, each with its own coefficient-range constraints. We compress only the public polynomial $\boldsymbol{h}$ and the ciphertext polynomial $\boldsymbol{c}$. For the secret polynomial $\boldsymbol{f}, \boldsymbol{f_2}$, we instead employ a direct split–join encoding to minimize computation time, at the expense of a slight increase in the private key storage. Table 8 details the encoding strategies for each polynomial type, and Table 9 summarizes the resulting storage requirements for all DAWN variants.

Table 8: Combination Strategy of DAWN

Scheme	Polynomial	N	u_j	$\lceil \log_2(p_j^{u_j}) \rceil$
DAWN-α-512	$\boldsymbol{h}$	769	5	48
	$\boldsymbol{c}$	110	5	34
DAWN-α-1024	$\boldsymbol{h}$	769	5	48
	$\boldsymbol{c}$	193	5	38
DAWN-β-512	$\boldsymbol{h}$	$258 = 3 \cdot 86$	(17, 7)	(27, 45)
	$\boldsymbol{c}$	$129 = 3 \cdot 43$	(17, 7)	(27, 38)
DAWN-β-1024	$\boldsymbol{h}$	$258 = 3 \cdot 86$	(17, 7)	(27, 45)
	$\boldsymbol{c}$	$258 = 3 \cdot 86$	(17, 7)	(27, 45)

6.3 Performance Benchmark

We give two implementations of our schemes: `REF` (plain `C`) and `AVX2`. All benchmarks were measured on a single thread of an Intel Core i9-11900K processor running at a frequency of 3.5GHz with TurboBoost and hyperthreading

Table 9: Storage for DAWN (in bytes)

Scheme	Ciphertext	Public Key	Private Key
DAWN-α-512	436	615	1319
DAWN-α-1024	973	1229	2605
DAWN-β-512	450	514	1154
DAWN-β-1024	1027	1027	2275

disabled. The compiler used is clang version 16.0.6 with the following compilation flags: `-O3, -march=native, -mtune=native, -fomit-frame-pointer`. The complete time cost data for DAWN are presented in Table 10, based on the average results from $1000,000$ experimental runs.

Table 10: Performance of DAWN (in cycles)

Scheme	Implementation	Key generation	Encapsulation	Decapsulation
DAWN-α-512	`REF`	92545	42128	45020
	`AVX2`	22571	20509	19975
DAWN-α-1024	`REF`	138860	77409	113827
	`AVX2`	46660	36816	38880
DAWN-β-512	`REF`	71173	35802	48495
	`AVX2`	23471	22336	23707
DAWN-β-1024	`REF`	135169	88260	127222
	`AVX2`	46997	42682	47987

7 Conclusion

We introduce DAWN, a NTRU-based encryption scheme that achieves smaller ciphertext and public key sizes among all known lattice-based KEMs while maintaining superior computational performance. Our zero divisor encoding technique and double encoding paradigm represent fundamental advances in NTRU design, transforming scheme optimization from ad hoc approaches into a principled framework. DAWN establishes new benchmarks for post-quantum encryption efficiency and addresses key practical barriers to post-quantum cryptography adoption. The significance of these results extends beyond incremental improvements. The double encoding paradigm provides a systematic approach for future NTRU developments, while an open problem remains: determining whether our polynomial choices for DAWN are optimal under the double encoding paradigm.

References

1. Alagic, G., Cooper, D., Dang, Q., Dang, T., Kelsey, J.M., Lichtinger, J., Liu, Y.K., Miller, C.A., Moody, D., Peralta, R., Perlner, R., Robinson, A., Smith-Tone, D., Apon, D.: Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process (2022-07-05 04:07:00 2022). https://doi.org/https://doi.org/10.6028/NIST.IR.8413
2. Albrecht, M.R., Cid, C., Faugère, J.C., Fitzpatrick, R., Perret, L.: Algebraic algorithms for LWE problems. ACM Commun. Comput. Algebra **49**(2), 62 (aug 2015). https://doi.org/10.1145/2815111.2815158
3. Albrecht, M.R., Cid, C., Faugère, J.C., Fitzpatrick, R., Perret, L.: On the complexity of the BKW algorithm on LWE. In: Budach, L. (ed.) Designs, Codes and Cryptography. pp. 325–354 (2015)
4. 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: Catalano, D., De Prisco, R. (eds.) Security and Cryptography for Networks. pp. 351–367. Springer International Publishing, Cham (2018)
5. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of Learning with Errors. Journal of Mathematical Cryptology **9**(3), 169–203 (2015). https://doi.org/doi:10.1515/jmc-2015-0016
6. Alkim, E., Cheng, D.Y.L., Chung, C.M.M., Evkan, H., Huang, L.W.L., Hwang, V., Li, C.L.T., Niederhagen, R., Shih, C.J., Wälde, J., Yang, B.Y.: Polynomial Multiplication in NTRU Prime: Comparison of Optimization Strategies on Cortex-M4 (2020). https://doi.org/10.46586/tches.v2021.i1.217-238
7. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange: a new hope. In: Proceedings of the 25th USENIX Conference on Security Symposium. p. 327–343. SEC'16, USENIX Association, USA (2016)
8. Arora, S., Ge, R.: New Algorithms for Learning in Presence of Errors. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) Automata, Languages and Programming. pp. 403–415. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
9. Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS–Kyber (2021)
10. Bai, S., Galbraith, S.D.: Lattice Decoding Attacks on Binary LWE. In: Susilo, W., Mu, Y. (eds.) Information Security and Privacy. pp. 322–337. Springer International Publishing, Cham (2014)
11. Bai, S., Jangir, H., Lin, H., Ngo, T., Wen, W., Zheng, J.: Compact Encryption Based on Module-NTRU Problems. In: Post-Quantum Cryptography: 15th International Workshop, PQCrypto 2024, Oxford, UK, June 12–14, 2024, Proceedings, Part I. p. 371–405. Springer-Verlag, Berlin, Heidelberg (2024). https://doi.org/10.1007/978-3-031-62743-9_13
12. Bernstein, D.J., Chuengsatiansup, C., Lange, T., van Vredendaal, C.: NTRU Prime: Reducing Attack Surface at Low Cost. In: Adams, C., Camenisch, J. (eds.) Selected Areas in Cryptography – SAC 2017. pp. 235–260. Springer International Publishing, Cham (2018)
13. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. J. ACM **50**(4), 506–519 (Jul 2003). https://doi.org/10.1145/792538.792543
14. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehle, D.: CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 353–367 (2018). https://doi.org/10.1109/EuroSP.2018.00032

15. Brakerski, Z., Döttling, N.: Lossiness and Entropic Hardness forRing-LWE. In: Pass, R., Pietrzak, K. (eds.) Theory of Cryptography. pp. 1–27. Springer International Publishing, Cham (2020)
16. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) Fully Homomorphic Encryption without Bootstrapping. ACM Trans. Comput. Theory **6**(3) (Jul 2014). https://doi.org/10.1145/2633600
17. Castryck, W., Iliashenko, I., Vercauteren, F.: Provably Weak Instances of Ring-LWE Revisited. In: Fischlin, M., Coron, J.S. (eds.) Advances in Cryptology – EUROCRYPT 2016. pp. 147–167. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
18. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic Encryption for Arithmetic of Approximate Numbers. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017. pp. 409–437. Springer International Publishing, Cham (2017)
19. Chuengsatiansup, C., Prest, T., Stehlé, D., Wallet, A., Xagawa, K.: Modfalcon: Compact signatures based on module-ntru lattices. In: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. p. 853–866. ASIA CCS '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3320269.3384758, https://doi.org/10.1145/3320269.3384758
20. Chung, C.M.M., Hwang, V., Kannwischer, M.J., Seiler, G., Shih, C.J., Yang, B.Y.: NTT Multiplication for NTT-unfriendly Rings: New Speed Records for Saber and NTRU on Cortex-M4 and AVX2. IACR Transactions on Cryptographic Hardware and Embedded Systems **2021**(2), 159–188 (Feb 2021). https://doi.org/10.46586/tches.v2021.i2.159-188
21. Cong Chen, Oussama Damba, J.r.H.s.A.H.J.R.J.M.S.P.S.W.W., Zhang, Z.: Ntru: Algorithm specifications and supporting documenta tion. Brown University and Onboard security company, Wilmington, USA (2019)
22. Ducas, L., Lyubashevsky, V., Prest, T.: Efficient Identity-Based Encryption over NTRU Lattices. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology – ASIACRYPT 2014. pp. 22–41. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
23. Ducas, L., van Woerden, W.: NTRU Fatigue: How Stretched is Overstretched? In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2021. pp. 3–32. Springer International Publishing, Cham (2021)
24. Duman, J., Hövelmanns, K., Kiltz, E., Lyubashevsky, V., Seiler, G., Unruh, D.: A Thorough Treatment of Highly-Efficient NTRU Instantiations. In: Public-Key Cryptography – PKC 2023: 26th IACR International Conference on Practice and Theory of Public-Key Cryptography, Atlanta, GA, USA, May 7–10, 2023, Proceedings, Part I. p. 65–94. Springer-Verlag, Berlin, Heidelberg (2023). https://doi.org/10.1007/978-3-031-31368-4_3
25. D'Anvers, J.P., Guo, Q., Johansson, T., Nilsson, A., Vercauteren, F., Verbauwhede, I.: Decryption Failure Attacks on IND-CCA Secure Lattice-Based Schemes. In: Public-Key Cryptography – PKC 2019. Lecture Notes in Computer Science, vol. 11443, pp. 565–598. Springer (2019). https://doi.org/10.1007/978-3-030-17259-6_19
26. Elias, Y., Lauter, K.E., Ozman, E., Stange, K.E.: Provably weak instances of Ring-LWE. CoRR **abs/1502.03708** (2015), http://arxiv.org/abs/1502.03708
27. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU (2019)

28. Fouque, P.A., Kirchner, P., Pornin, T., Yu, Y.: BAT: Small and Fast KEM over NTRU Lattices. IACR Transactions on Cryptographic Hardware and Embedded Systems **2022**, 240–265 (02 2022). https://doi.org/10.46586/tches.v2022.i2.240-265
29. Guo, Q., Johansson, T.: Faster Dual Lattice Attacks for Solving LWE with Applications to CRYSTALS. In: Advances in Cryptology – ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part IV. p. 33–62. Springer-Verlag, Berlin, Heidelberg (2021). https://doi.org/10.1007/978-3-030-92068-5_2
30. Guo, Q., Johansson, T., Yang, J.: A Novel CCA Attack Using Decryption Errors Against LAC. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology – ASIACRYPT 2019. pp. 82–111. Springer International Publishing, Cham (2019)
31. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: Buhler, J.P. (ed.) Algorithmic Number Theory. pp. 267–288. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)
32. Hoffstein, J., Silverman, J.: Optimizations for NTRU. In: Alster, K., Urbanowicz, J., Williams, H.C. (eds.) Public-Key Cryptography and Computational Number Theory. pp. 77–88. De Gruyter, Berlin, New York (2001). https://doi.org/doi:10.1515/9783110881035.77
33. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A Modular Analysis of the Fujisaki-Okamoto Transformation. In: TCC (1). Lecture Notes in Computer Science, vol. 10677, pp. 341–371. Springer (2017)
34. Hülsing, A., Rijneveld, J., Schanck, J., Schwabe, P.: High-Speed Key Encapsulation from NTRU. In: Fischer, W., Homma, N. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2017. pp. 232–252. Springer International Publishing, Cham (2017)
35. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: IND-CCA-Secure Key Encapsulation Mechanism in the Quantum Random Oracle Model, Revisited. In: CRYPTO (3). Lecture Notes in Computer Science, vol. 10993, pp. 96–125. Springer (2018)
36. Kannan, R.: Minkowski's Convex Body Theorem and Integer Programming. Mathematics of Operations Research **12**(3), 415–440 (1987)
37. Kim, J., Park, J.H.: NTRU+: Compact Construction of NTRU Using Simple Encoding Method. IEEE Transactions on Information Forensics and Security **18**, 4760–4774 (2023). https://doi.org/10.1109/TIFS.2023.3299172
38. Kirchner, P., Espitau, T., Fouque, P.A.: Fast Reduction of Algebraic Lattices over Cyclotomic Fields. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020. pp. 155–185. Springer International Publishing, Cham (2020)
39. Liang, Z., Fang, B., Zheng, J., Zhao, Y.: Compact and Efficient KEMs over NTRU Lattices (2022), https://arxiv.org/abs/2205.05413
40. López-Alt, A., Tromer, E., Vaikuntanathan, V.: On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing. p. 1219–1234. STOC '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2213977.2214086
41. Lu, X., Liu, Y., Zhang, Z., Jia, D., Xue, H., He, J., Li, B., Wang, K.: LAC: Practical Ring-LWE Based Public-Key Encryption with Byte-Level Modulus. Cryptology ePrint Archive, Paper 2018/1009 (2018), https://eprint.iacr.org/2018/1009
42. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors over Rings. In: Gilbert, H. (ed.) Advances in Cryptology – EUROCRYPT 2010. pp. 1–23. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)

43. Lyubashevsky, V., Seiler, G.: NTTRU: Truly Fast NTRU Using NTT. IACR Transactions on Cryptographic Hardware and Embedded Systems **2019**(3), 180–201 (May 2019). https://doi.org/10.13154/tches.v2019.i3.180-201
44. MATZOV: Report on the Security of LWE: Improved Dual Lattice Attack (2022), https://api.semanticscholar.org/CorpusID:251600824
45. NIST: NIST: Post-Quantum Cryptography Standardization (2017), https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/submission-requirements
46. Pellet-Mary, A., Stehlé, D.: On the Hardness of the NTRU Problem. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2021. pp. 3–35. Springer International Publishing, Cham (2021)
47. Schnorr, C.P., Euchner, M.: Lattice basis reduction: Improved practical algorithms and solving subset sum problems. In: Budach, L. (ed.) Fundamentals of Computation Theory. pp. 68–85. Springer Berlin Heidelberg, Berlin, Heidelberg (1991)
48. Seiler, G.: Faster AVX2 optimized NTT multiplication for Ring-LWE lattice cryptography (2018), report 2018/039
49. Stehlé, D., Steinfeld, R.: Making NTRU as Secure as Worst-Case Problems over Ideal Lattices. In: Paterson, K.G. (ed.) Advances in Cryptology – EUROCRYPT 2011. pp. 27–47. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
50. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient Public Key Encryption Based on Ideal Lattices. In: Matsui, M. (ed.) Advances in Cryptology – ASIACRYPT 2009. pp. 617–635. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
51. Zhang, J., Feng, D., Yan, D.: NEV: Faster and Smaller NTRU Encryption Using Vector Decoding. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023. pp. 157–189. Springer Nature Singapore, Singapore (2023)