Almost NTRU: Revisiting Noncommutativity Against Lattice Attacks*

Ali Raya[1], Vikas Kumar[2], Seong Oun Hwang[2], and Sugata Gangopadhyay[1]

[1] Department of Computer Science and Engineering
Indian Institute of Technology Roorkee, Haridwar 247667, India.
{ali_r, sugata.gangopadhyay}@cs.iitr.ac.in
[2] Department of Computer Engineering
Gachon University, Seongnam 13307, South Korea.
{vikaskumar250697, sohwang}@gachon.ac.kr

Abstract. NTRU is one of the most extensively studied lattice-based cryptographic schemes and is widely regarded as a strong candidate for post-quantum security. The most effective attacks on NTRU are lattice-based or lattice-related, which naturally guide the choice of parameters to achieve the desired security levels. In 1997, Hoffstein and Silverman proposed a variant of NTRU based on a noncommutative algebraic structure, claiming that it would mitigate lattice attacks. However, their scheme was later shown to be vulnerable to an algebraic attack by Coppersmith. Although several subsequent attempts have been made in the literature to develop noncommutative variants of NTRU, most of these designs have either been shown to be vulnerable to algebraic attacks or have failed to directly address lattice-based attacks.

In this work, we revisit the problem of constructing a noncommutative analog of NTRU that offers stronger resistance against direct lattice attacks. Firstly, we conceptualize the problem by introducing an almost unstructured variant, and then refine this idea towards a more compact instantiation, culminating in a fully structured construction defined over the group ring of the dihedral group. Our proposal may be viewed as a follow-up to the early noncommutative construction of Hoffstein and Silverman.

We further provide a complete reference implementation of the structured construction under two proposed parameter sets, *Plausible* and *Paranoid*, demonstrating both the efficiency and compactness of our scheme in comparison with NTRU-HPS and the state-of-the-art noncommutative NTRU variant.

Keywords: Post-Quantum · NTRU · Lattice attacks · Noncommutative

* As an initial step toward evaluating the hardness of the ANTRU assumption, we have formulated it as a problem in the NSUCRYPTO 2025 Olympiad https://nsucrypto.nsu.ru/olymp/2025/

1 Introduction

NTRU is one of the earliest proposals for a post-quantum cryptographic scheme, conjectured to be secure based on hard mathematical problems related to lattices. The original construction of NTRU was introduced by Hoffstein, Pipher, and Silverman in 1997, based on a commutative ring of truncated polynomials. Although the original NTRU assumption was not initially grounded in formal security proofs, its credibility has been reinforced over nearly three decades of cryptanalysis. This record of cryptanalysis has led to the development of versatile cryptographic primitives and advanced constructions whose security is based on the NTRU assumption. The flexibility in designing NTRU-like schemes has led to numerous variants defined over different algebraic rings, motivated by both security considerations and performance improvements. The original formulation of NTRU was introduced over the truncated polynomial ring, where the task of recovering the private key is reduced to finding a short vector (corresponding to the private key) in the lattice associated with the construction. Since then, the most effective attacks have consistently been based on lattice techniques, either in isolation or in combination with other methods. Naturally, lattice attacks have significantly influenced the parameter selection of NTRU-like constructions. With advances in cryptanalytic techniques and lattice reduction algorithms, parameters have been re-evaluated to require larger dimensions, which in turn lead to increased ciphertext sizes, larger public keys, and higher computational costs. Although the idea of employing noncommutative variants of NTRU to mitigate the impact of lattice attacks was already hinted at in the early cryptanalysis of Coppersmith and Shamir [10], to the best of our knowledge, only one concrete attempt—by Hoffstein and Silverman [23], which ultimately proved unsuccessful—has explored a noncommutative construction as a means of resistance. Other works considering noncommutative settings have merely instantiated NTRU over alternative rings, which do not fundamentally prevent lattice attacks and, in some cases, yield weaker instances. Hence, in our view, it is worthwhile to revisit the idea of designing a genuinely noncommutative NTRU construction—one that remains faithful to the spirit of NTRU while offering a promising path towards more compact schemes that either resist direct lattice attacks or mitigate their impact on parameter selection.

Many Variants. Following the original proposal of NTRU [22], numerous variants have been developed by instantiating the NTRU assumption over different algebraic structures. Most of the prominent designs have been defined over commutative algebraic rings. Among them, NTRU-HPS and NTRU-HRSS [27], submitted jointly under the name NTRU, advanced to the third round of the NIST standardization process as finalist candidates. In parallel, NTRU Prime [5] reached the same round as an alternate candidate. It is worth noting, however, that these constructions were ultimately outperformed by the selected standard Kyber [44], largely because the NTRU-based submissions were not designed over Number Theoretic Transform (NTT)-friendly rings, limiting their implementation efficiency. To address this limitation, NTTRU [37] was later introduced by Lyubashevsky and Seiler as an efficient Key Encapsulation Mechanism (KEM)

based on an NTT-friendly cyclotomic ring, achieving significantly better performance compared to the NTRU KEMs submitted to NIST. Building on this idea, NTRU+ was proposed and submitted to the Korean Post-Quantum Cryptography competition (KpqC). In 2025, NTRU+ [30] was selected for standardization by the KpqC as an efficient analog of NTRU.

On the other hand, the literature has explored a line of constructions based on noncommutative algebra, aiming to strengthen the security of NTRU against algebraic attacks or to increase its overall resilience. The first such work, proposed by Hoffstein and Silverman [23], introduced the use of noncommutativity as a means to avoid lattice-based attacks. However, the design deviated significantly from the original NTRU structure in its encryption and decryption procedures, which eventually led to a successful cryptanalysis [9,46] that rendered the scheme insecure. Following this, several proposals [42,41,33,38,3,47] introduced NTRU over various noncommutative structures. Nevertheless, none of these constructions managed to fully resist direct lattice attacks, resulting in performance and security levels comparable to standard NTRU defined over commutative rings. Moreover, some proposals relied on highly structured rings, which in turn enabled even stronger lattice-based attacks, such as the one demonstrated by Raya et al. [40].

Context. The original NTRU cryptosystem was introduced as "a mixing system based on polynomial algebra and reduction modulo." During encryption, the message is mixed with a polynomial from the ring in such a way that only the party possessing the trapdoor associated with the public key of the cryptosystem can later *unmix* it. The public key equation of NTRU over the commutative truncated polynomial ring is given as

$$h = f^{-1} * g \pmod{q}, \tag{1}$$

where both f and g are chosen with small Euclidean norms and form the trapdoor associated with the public key.

On the other hand, the key equation of our construction, Almost NTRU (ANTRU), can be viewed as a rearrangement of the standard NTRU key equation, but formulated over a noncommutative ring rather than a commutative one, as in most conventional NTRU constructions.

In a commutative ring, decomposing f as $f = f_1 * f_2 \pmod{q}$ and rewriting the public key equation as

$$h = f_1^{-1} * g * f_2^{-1} \pmod{q}$$

has no effect, since multiplication commutes. However, in the noncommutative setting, this no longer holds:

$$h' = f_1^{-1} * g * f_2^{-1} \pmod{q} \neq h.$$

Consequently, the key equation of ANTRU is formulated in terms of h', where f_1 and f_2 are restricted to a commutative subring (to enable decryption), while g is sampled from the noncommutative ring (to resist direct lattice attacks). It is

useful to view ANTRU as a two-layer NTRU problem: $f_1^{-1} * g' \pmod q$, where g' itself is defined as another NTRU instance, $g' = g * f_2^{-1} \pmod q$.

The idea of introducing a two-layer cryptographic hardness assumption has appeared previously, for example, in NTWE [18], where the authors modify the NTRU key generation equation $f^{-1} * g \pmod q$ by letting g be an LWE or Module-LWE sample. Interestingly, this resulted in more compact parameter sets than those of pure Module-LWE constructions. In contrast, our construction of ANTRU employs two layers of NTRU instances — a phenomenon that cannot arise in the commutative setting.

Our Work

- **Conceptualization of ANTRU:** Different from previous NTRU variants defined over noncommutative algebra, this work introduces a framework for constructing a noncommutative analog of NTRU in which direct lattice attacks alone appear ineffective for both key-recovery and message-recovery. We formulate the decision and search versions of the *Almost NTRU* problem (abbreviated as ANTRU) and describe the general attack landscape, supported by lower-bound security guarantees inherited from the NTRU assumption for the key generation. To concretize ANTRU, we present three illustrative instantiations: ① Almost-unstructured, ② Semi-structured, and ③ Fully-structured, with a special emphasis on the fully structured variant.

- **Fully Structured Variant:** The fully structured variant of ANTRU can be viewed as a repair to the early Hoffstein-Silverman [23] design, which was broken by Coppersmith [9] and subsequently studied in detail by Truman [46]. Although the Hoffstein-Silverman construction operates over the group ring $\mathbb{Z}D_N$ (with $D_N = \langle x, y \mid x^N = 1,\ y^2 = 1,\ xy = yx^{N-1} \rangle$), and is therefore similar in ambient algebra to our construction, our scheme differs significantly from Hoffstein-Silverman and is closer in spirit to the NTRU-HPS design [22]. The most relevant differences and security considerations are as follows:
  - **Key Generation.** In the Hoffstein-Silverman construction the public key is formed as $h = f_1 * g * f_1^{-1} \pmod q$, where f_1 is chosen from a (large) commutative subring $\mathcal{S} = \{ f \in \mathbb{Z}D_N \mid fy = yf \}$, i.e., elements that commute with the reflection y. Suppose an adversary can devise a special-purpose map θ that effectively transforms the non-commutative conjugation into a multiplication by an element of $\mathcal{S}$; for example,

    $$\theta(f_1 * g * f_1^{-1}) \ = \ f_1 * \theta(g) * f_1^{-1},$$

    for some $\theta(g) \in \mathcal{S}$. If such a map exists, then applying θ to the public key together with the relation $f_1 * f_1^{-1} = 1 \pmod q$ allows the adversary to recover $\theta(g)$. Knowledge of $\theta(g)$ (an element of the commutative subring $\mathcal{S}$) can substantially simplify decryption or otherwise leak structure that enables message recovery without explicitly recovering f_1.

In contrast, our key generation is defined as $h = f_1^{-1} * g * f_2^{-1}$, where the product $f_1 * f_2 \in \mathcal{S}$ is chosen to be relatively small in Euclidean norm. Consequently, any special-purpose map (if it exists) would transform the key equation into an *NTRU-type instance*, rather than directly revealing information about the secret element g. However, to further mitigate the impact of such a map, and therefore mitigate the lattice-related attacks, we modify the sampling space of g, as described in the next point.

- **Sampling Space.** Since the group element y in the dihedral group belongs to $\mathcal{S}$, it is easy to verify that the map $\theta : h \mapsto h + yhy$, when applied to the public key of the Hoffstein-Silverman scheme, directly yields $g + ygy$. In contrast, when applied to the public key of our proposed scheme, it produces an *NTRU-type instance* of the form $(f_1 * f_2)^{-1} * (g + ygy) \pmod q$, which requires solving the NTRU assumption over the underlying ring to extract any useful information that may indirectly assist in decryption. To further mitigate the possibility of such a map being used to construct a distinguisher against the public key (at a cost comparable to applying lattice reduction to the lattice associated with $\theta(h) = h + yhy$), we restrict g to be sampled from the kernel of θ, i.e.,
$$\mathcal{S}^- \;=\; \{f \in \mathbb{Z}D_N \mid fy = -yf\},$$
which remains sufficiently large in size (depending on N, the parameter defining the dihedral group). Consequently, the public key in our scheme takes the form $h = f_1^{-1} * g * f_2^{-1} \pmod q$ with $h \in \mathcal{S}^-$. Hence, finding a distinguisher against the public key becomes significantly harder, since the map $\theta : h \mapsto h + yhy$ (and related maps) no longer yields an NTRU-type instance, but instead the map θ evaluates to 0 for every element of $\mathcal{S}^-$, the set to which our public key belongs.
Further, for the dihedral group of order $2N$, any element $a = a_0(x) + y\,a_1(x) \in \mathcal{S}$ (or $\mathcal{S}^-$) is uniquely determined by the first $\lfloor N/2 \rfloor + 1$ coefficients of both $a_0(x)$ and $a_1(x)$. Consequently, every multiplication in our cryptosystem between elements of $\mathcal{S}$ and $\mathcal{S}^-$ (in either order) can be realized by computing only the $2\lfloor N/2 \rfloor + 2$ independent coefficients of the product, as the remaining coefficients are either mirrored (for results in $\mathcal{S}$) or negated mirrors (for results in $\mathcal{S}^-$). Intuitively, this reduces the lattice dimension relevant to potential lattice-based attacks (when applicable) from $4N$ to $2N$, while simultaneously allowing larger values of N to be chosen without performance penalties, since the structure supports faster multiplication and a more compact public-key representation, requiring only the unique coefficients to be computed and stored.
- **Twisting Multiplication.** Although direct lattice attacks are not the most effective against the structured variant, we replace the group ring of the dihedral group with the twisted group ring (Definition 6). The lattice associated with this twisted definition eliminates unnecessary homomorphisms that could otherwise weaken the hardness of the underlying lattice problem. At a higher level of abstraction, the lattice associated with the twisted dihedral group ring is analogous to the lattice arising from the ring $\mathbb{Z}[x]/(x^n + 1)$, which is preferred over $\mathbb{Z}[x]/(x^n - 1)$ for

composite n, since it avoids vulnerabilities to attacks of the type introduced by Gentry against composite n [20].

– **Parameter Selection and Reference Implementation:** Following our discussion on the concrete security estimation for ANTRU, we parametrize the structured variant using two approaches: ① *plausible* and ② *paranoid*. In the plausible scenario, we choose parameters by ruling out the possibility of an effective lattice attack against the construction, as supported by the discussion in Section 4. In the paranoid scenario, we conservatively assume an adversary can somehow derive a mapping that converts the noncommutative public-key equation into an NTRU-like form; hence, the security estimates for this case are driven by lattice attacks. Although our security analysis considers the paranoid scenario highly unlikely, we nevertheless include parameter sets for it to illustrate an extremely conservative security estimate. Table 1 compares the computational and bandwidth requirements of our construction with the parameter sets selected for NTRU-HPS from [7] and with the noncommutative variant of NTRU from [42]. We choose these two reference points because (i) comparison with NTRU-HPS positions our scheme relative to well-known NTRU/LWE constructions, and (ii) the noncommutative variant of [42] represents the best performance/compactness among several noncommutative NTRU proposals [41,33]. The parameter sets are organized into three groups with security levels comparable to those introduced in NTRU-HPS [7]. The column labeled **"Dim"** denotes the minimum dimension of the coefficient vector corresponding to the public key, while $\mathbf{log_2(}q\mathbf{)}$ indicates the base-2 logarithm of the modulus. The column **"CT/PK"** refers to the size of the ciphertext or public key in bytes. Meanwhile, the column $\boldsymbol{\beta}$ specifies the blocksize corresponding to the primal attack, and the column **"Estimate"** provides the classical security estimate given as (primal attack cost, hybrid attack cost) when lattice attacks are considered, or as the search-related cost when lattice attacks are deemed ineffective. The rightmost part of the table reports the reference implementation performance in terms of CPU cycles for key generation, encapsulation, and decapsulation, respectively, measured on a single core of the 13$^{\text{th}}$ Generation Intel Core i7-13700 processor, operating at a frequency of 1.5 GHz per core. The system is equipped with 32 GiB of RAM, a 640 KiB L1 data cache, a 768 KiB L1 instruction cache, a 24 MiB L2 cache, and a 30 MiB L3 cache. Measurements were performed on Ubuntu 22.04.5 LTS (release 22.04), with Turbo Boost and hyper-threading disabled. The compiler used was GCC (Ubuntu 11.4.0-1ubuntu1,22.04.2), with **no optimization** flag enabled.
It is worth mentioning that although the lattice dimension—where lattice-related attacks are estimated—is smaller for ANTRU and the corresponding lattice volume is larger, the norm of the target secret $(f_1 * f_2, g)$ is significantly higher, since $f_1 * f_2$ is not a ternary vector as in the other compared NTRU constructions. Consequently, lattice estimators yield larger blocksizes even for smaller lattice dimensions. Furthermore, the ANTRU parameter sets do not fall within the overstretched regime (detailed discussion is

given in Subsection 5.2). From the results presented in Table 1, it is evident that ANTRU (*plausible* and *paranoid*) exhibits significantly more compact ciphertext and public key sizes compared to both NTRU and DiTRU$^+$. Moreover, the performance of ANTRU (*plausible*) is more efficient in encapsulation and decapsulation than both NTRU and DiTRU$^+$. Additionally, ANTRU demonstrates a considerably faster key generation process compared to NTRU across all parameter sets, and is comparably faster than DiTRU$^+$.

Table 1: Comparison of the parameter sets of ANTRU (this work), NTRU-HPS [7], and the noncommutative variant of NTRU [42]. The comparison is organized into three groups according to security levels, following the categories introduced for NTRU-HPS.

						Ref Implementation (#CPU cycles $\times 10^3$)		
	Dim	**$\log_2(q)$**	**CT/PK**	**β**	**Estimate**	**KeyGen**	**Encap**	**Decap**
ntru-hps_2048509	509	11	630	364	(106, 105)	54 843	2 379	5 796
DiTRU$^+$_2048269	2×269	11	737	401	(117, 111)	28 920	2 681	6 542
ANTRU_8192269 [plausible]	269	13	436	-	> 128	24 883	1 887	4 200
ANTRU_8192379 [paranoid]	379	13	615	386	(113, 105)	48 838	3 318	7 902
ntru-hps_2048677	677	11	930	496	(145, 144)	101 348	3 991	10 172
DiTRU$^+$_2048347	2×347	11	952	540	(158, 147)	47 945	4 277	10 759
ANTRU_16384293 [plausible]	293	14	511	-	> 192	27 549	2 183	4 927
ANTRU_16384509 [paranoid]	509	14	889	534	(156, 157)	80 673	5 527	13 872
ntru-hps_4096821	821	12	1230	612	(179, 178)	138 782	5 705	14 883
DiTRU$^+$_4096419	2×419	12	1254	637	(186, 182)	70 986	6 089	15 678
ANTRU_16384509 [plausible]	509	14	889	-	> 256	82 256	5 531	13 850
ANTRU_16384653 [paranoid]	653	14	1141	680	(199, 180)	134 592	8 674	22 534

The reference implementation and accompanying artifacts are publicly available at `https://github.com/The-Isogeniest/ANTRU`.

1.1 Organization

The remainder of this paper is structured as follows. Section 2 establishes the necessary background on lattices, NTRU, and group rings. Section 3 introduces our three proposed variants of noncommutative NTRU. We then analyze the security of the fully structured variant in Section 4 and discuss the parameter selection in Section 5. Finally, Section 6 concludes the paper and outlines promising avenues for future research. Supplementary details are provided in the Appendices.

2 Preliminaries

2.1 Notations

We use '$*$' to denote multiplication between two elements in their underlying algebraic ring. The sets $\mathbb{Z}$ and $\mathbb{R}$ denote the integers and real numbers, respectively. Let $\mathbb{Z}_q$ denote the ring of integers modulo q, and let $\mathbb{Z}_q^\times$ be its group of units. The symbol $\mathcal{R}$ denotes an arbitrary ring or otherwise explicitly defined,

and $M_n(\mathcal{R})$ denotes the set of all $n \times n$ matrices with entries from $\mathcal{R}$. Vectors are represented by bold lowercase letters, while matrices are denoted by bold uppercase letters. For a given matrix $\mathbf{X}$, we use the following notations: $\mathbf{X}^{(i,j)}$ denotes the element at row i and column j; $\mathbf{X}^{(i,:)}$ denotes the i-th row; $\mathbf{X}^{(:,j)}$ denotes the j-th column; $\mathbf{X}^{Tr}$ denotes the transpose of $\mathbf{X}$; and $\overrightarrow{\mathbf{X}}$ denotes the vectorization of $\mathbf{X}$ as a single column vector, obtained by stacking the rows of $\mathbf{X}$ from top to bottom. For a polynomial $v(x) = v_0 + v_1 x + \ldots + v_{n-1}x^{n-1}$ with coefficients $v_i \in \mathbb{Z}$, we define the vector $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1}) \in \mathbb{Z}^n$ as the coefficient vector of the polynomial. The Euclidean norm of $\mathbf{v}$ is defined as $\|\mathbf{v}\| = \sqrt{\sum_{i=0}^{n-1} v_i^2}$, the supremum norm is defined as $\|v\|_\infty = \max_i |v_i|$, and the inner product between two vectors $\mathbf{u} = (u_0, u_1, \ldots, u_{n-1})$ and $\mathbf{v}$ is given by $\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=0}^{n-1} u_i v_i$. For $m \geq 1$, $\mathbf{I}_m$ and $\mathbf{0}_m$ denote identity and zero matrices of order m, respectively, and 0_m represents the zero vector of length m. Suppose D is a distribution on a set S, then $s \leftarrow D$ denotes that $s \in S$ is sampled according to the distribution D. The notation $U(S)$ represents a uniform distribution over a finite set S. For uniform distribution, we simply use $s \leftarrow_\$ S$ to denote that s is sampled uniformly at random from S. $O(\cdot), o(\cdot)$, and $\Theta(\cdot)$ are usual Big-O, small-o, and Big-Theta notations, respectively.

This work assumes familiarity with standard cryptographic terminologies and security notions, including Public-Key Encryption (PKE), Key-Encapsulation Mechanism (KEM), One-Wayness under Chosen-Plaintext Attack (OW-CPA), Indistinguishability under Chosen-Plaintext Attack (IND-CPA), and Indistinguishability under Chosen-Ciphertext Attack (IND-CCA). Due to page limitations, the formal definitions are provided in the Supplementary material A.

2.2 Lattices

Definition 1 (Lattice). *Let* $\mathbf{B} = \left(\mathbf{b}_0 | \mathbf{b}_1 | \ldots | \mathbf{b}_{n-1}\right) \in \mathbb{Z}^{n \times d}$ *be a matrix whose rows* $\mathbf{b}_i \in \mathbb{Z}^d$ *are linearly independent. The lattice generated by* $\mathbf{B}$ *is defined as*

$$\mathcal{L}(\mathbf{B}) = \{\mathbf{x}\mathbf{B} \mid \mathbf{x} \in \mathbb{Z}^n\},$$

i.e., the set of all integer linear combinations of the rows of $\mathbf{B}$*.*

Here, n is the rank of the lattice, and d is its dimension. The lattice is called full-rank if $n = d$, in which case the volume of the lattice is given by $\mathsf{vol}(\mathcal{L}(\mathbf{B})) = |\det(\mathbf{B})|$. In this work, we consider only full-rank lattices.

Definition 2 (Shortest Vector Problem (SVP)). *Given a full-rank lattice* $\mathcal{L}(\mathbf{B})$ *generated by the basis* $\mathbf{B}$*, the* Shortest Vector Problem (SVP) *asks to find a non-zero vector* $\mathbf{v} \in \mathcal{L}(\mathbf{B}) \setminus \{\mathbf{0}\}$ *such that* $\|\mathbf{v}\| = \min_{\substack{\mathbf{w} \in \mathcal{L}(\mathbf{B}) \\ \mathbf{w} \neq \mathbf{0}}} \|\mathbf{w}\|$*. The minimum norm of a non-zero lattice vector is called the first successive minimum and denoted by* $\lambda_1(\mathcal{L}(\mathbf{B}))$*.*

Definition 3 (Closest Vector Problem (CVP)). *Given a full-rank lattice* $\mathcal{L}(\mathbf{B})$ *of dimension* n*, and a vector* $\mathbf{t} \in \mathbb{R}^n$ *(does not lie in the lattice), the* Closest

Vector Problem (CVP) *asks to find* $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ *which is the closest vector to* $\mathbf{t}$ *i.e.,* $\|\mathbf{v} - \mathbf{t}\| \leq \|\mathbf{w} - \mathbf{t}\|$ *for all* $\mathbf{w} \in \mathcal{L}(\mathbf{B})$.

The SVP and the CVP are computationally hard for certain types of bases, referred to as bad bases. This hardness forms the foundational assumption for the security of many cryptosystems in lattice-based cryptography. Furthermore, using *Kannan's embedding* [29], one can reduce a CVP instance in dimension n with respect to a target vector $\mathbf{t} \in \mathbb{R}^n$ to an SVP instance in dimension $n+1$ by constructing the following lattice basis:

$$\mathbf{B}' = \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{t} & u \end{pmatrix},$$

where u is the embedding factor, typically chosen as $u = 1$.

Definition 4 (Gaussian Heuristic). *The expected norm of the shortest non-zero vector* $\lambda_1(\mathcal{L}(\mathbf{B}))$ *in an* n*-dimensional lattice, according to the Gaussian heuristic, is denoted by* $\mathsf{gh}(\mathcal{L}(\mathbf{B}))$ *and is given by*

$$\mathsf{gh}(\mathcal{L}(\mathbf{B})) = \sqrt{\frac{n}{2\pi e}} \cdot \mathsf{vol}(\mathcal{L}(\mathbf{B}))^{1/n} \tag{2}$$

2.3 Group Ring

Definition 5 (Group Ring). *For a group* $\mathcal{G} = \{g_0, g_1, \ldots, g_{n-1}\}$ *of finite order* n *and a ring* $\mathcal{R}$*, the group ring* $\mathcal{R}[\mathcal{G}]$ *is defined as the set of all formal sums:*

$$\mathcal{R}[\mathcal{G}] = \left\{ a = \sum_{i=0}^{n-1} \alpha_{g_i} g_i \ : \ \alpha_{g_i} \in \mathcal{R} \right\}. \tag{3}$$

Let $x = \sum_{i=0}^{n-1} \alpha_{g_i} g_i,\ y = \sum_{i=0}^{n-1} \beta_{g_i} g_i$ be two elements in $\mathcal{R}[\mathcal{G}]$. Then, $\mathcal{R}[\mathcal{G}]$ forms a ring under the following operations:

- **Addition:** $x + y = \sum_{i=0}^{n-1} (\alpha_{g_i} + \beta_{g_i}) g_i$
- **Multiplication:** $x * y = \sum_{k=0}^{n-1} \gamma_{g_k} g_k$, where $\gamma_{g_k} = \sum_{g_i g_j = g_k} \alpha_{g_i} \beta_{g_j}$.

If $\mathcal{R}$ has a multiplicative identity $1_{\mathcal{R}}$, then the group ring $\mathcal{R}[\mathcal{G}]$ is a ring with unity given by

- **Unity:** $1_{\mathcal{R}[\mathcal{G}]} = \sum_{g \in \mathcal{G}} a_g g$, where $a_e = 1_{\mathcal{R}}$ and $a_g = 0$ for $g \neq e$,

with e denoting the identity element of the group $\mathcal{G}$. Scalar multiplication can be defined for $\lambda \in \mathcal{R}$ and an element $x = \sum_{i=0}^{n-1} \alpha_{g_i} g_i \in \mathcal{R}[\mathcal{G}]$ as

- **Scalar multiplication:** $\lambda x = \lambda \left(\sum_{i=0}^{n-1} \alpha_{g_i} g_i \right) = \sum_{i=0}^{n-1} (\lambda \alpha_{g_i}) g_i$.

Additionally, if $\mathcal{R}$ is commutative, then $\mathcal{R}[\mathcal{G}]$ is naturally an $\mathcal{R}$-algebra. We can interchangeably use the vector form to represent an element $x \in \mathcal{R}[\mathcal{G}]$ as

- **Coefficient vector:** $\mathbf{x} = (\alpha_{g_0}, \alpha_{g_1}, \ldots, \alpha_{g_{n-1}}) \in \mathcal{R}^n$,

where the coefficient α_{g_i} corresponds to the group element $g_i \in \mathcal{G}$.

Matrix Representation: In [28], the author defines an isomorphism between the group ring $\mathcal{R}[\mathcal{G}]$ and a certain subring of $n \times n$ matrices over $\mathcal{R}$. The map τ sends an element $x \in \mathcal{R}[\mathcal{G}]$ to its corresponding matrix $\mathcal{M}_{\mathcal{R}[\mathcal{G}]}(x) \in \mathcal{M}_n(\mathcal{R})$, defined as follows:

$$\begin{aligned} &\tau : x \mapsto \mathcal{M}_{\mathcal{R}[\mathcal{G}]}(x) \\ &\sum_{i=0}^{n-1} \alpha_{g_i} g_i \mapsto \begin{pmatrix} \alpha_{g_0^{-1}g_0} & \alpha_{g_0^{-1}g_1} & \cdots\cdots & \alpha_{g_0^{-1}g_{n-1}} \\ \alpha_{g_1^{-1}g_0} & \alpha_{g_1^{-1}g_1} & \cdots\cdots & \alpha_{g_1^{-1}g_{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{g_{n-1}^{-1}g_0} & \alpha_{g_{n-1}^{-1}g_1} & \cdots\cdots & \alpha_{g_{n-1}^{-1}g_{n-1}} \end{pmatrix}. \end{aligned} \tag{4}$$

Let $\mathbf{x} = (\alpha_{g_0}, \alpha_{g_1}, \ldots, \alpha_{g_{n-1}})$ and $\mathbf{y} = (\beta_{g_0}, \beta_{g_1}, \ldots, \beta_{g_{n-1}})$ be the vectors corresponding to two elements $x, y \in \mathcal{R}[\mathcal{G}]$. Then, the following results hold:

1. Vector addition corresponds to component-wise addition:

$$\mathbf{x} + \mathbf{y} = (\alpha_{g_0} + \beta_{g_0}, \alpha_{g_1} + \beta_{g_1}, \ldots, \alpha_{g_{n-1}} + \beta_{g_{n-1}}).$$

2. Vector multiplication corresponds to the multiplication of $\mathbf{x}$ with the matrix representation of y:

$$\mathbf{x} * \mathbf{y} = \mathbf{x}\mathcal{M}_{\mathcal{R}[\mathcal{G}]}(y) = (\alpha_{g_0}, \alpha_{g_1}, \ldots, \alpha_{g_{n-1}}) \begin{pmatrix} \beta_{g_0^{-1}g_0} & \beta_{g_0^{-1}g_1} & \cdots\cdots & \beta_{g_0^{-1}g_{n-1}} \\ \beta_{g_1^{-1}g_0} & \beta_{g_1^{-1}g_1} & \cdots\cdots & \beta_{g_1^{-1}g_{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{g_{n-1}^{-1}g_0} & \beta_{g_{n-1}^{-1}g_1} & \cdots\cdots & \beta_{g_{n-1}^{-1}g_{n-1}} \end{pmatrix}.$$

3. The map τ is a bijective ring homomorphism [28, Theorem 1]. Specifically, $\tau(x + y) = \tau(x) + \tau(y)$ and $\tau(x * y) = \tau(x)\tau(y)$.
4. If $\mathcal{G}$ is abelian and $\mathcal{R}$ is a commutative ring, then $\tau(x)\tau(y) = \tau(y)\tau(x)$, and vector–matrix multiplication also commutes: $\mathbf{x} * \mathbf{y} = \mathbf{x}\tau(y) = \mathbf{y}\tau(x)$.

Definition 6 (Twisted Group Ring). *The twisted group ring $\mathcal{R}^{\lambda}[\mathcal{G}]$ is defined similarly to the group ring in Definition 3, except that the multiplication is* twisted *according to a certain map λ:*

$$x * y = \sum_{k=0}^{n-1} \gamma_{g_k}^{\lambda} g_k, \quad \textit{where} \quad \gamma_{g_k}^{\lambda} = \sum_{g_i g_j = g_k} \lambda(g_i, g_j)\alpha_{g_i}\beta_{g_j}, \tag{5}$$

where $\lambda : \mathcal{G} \times \mathcal{G} \longrightarrow \mathbb{R}^{\times}$ is called a 2-cocycle that satisfies the following properties:

1. $\lambda(1, 1) = 1$.
2. $\lambda(g_0 g_1, g_2)\,\lambda(g_0, g_1) = \lambda(g_0, g_1 g_2)\,\lambda(g_1, g_2)$ *for all* $g_0, g_1, g_2 \in \mathcal{G}$.

The matrix representation of $x \in \mathcal{R}^{\lambda}[\mathcal{G}]$ is given according to the map $\tau^{\lambda} : x \mapsto \mathcal{M}_{\mathcal{R}^{\lambda}[\mathcal{G}]}(x)$, where

$$\mathcal{M}_{\mathcal{R}^{\lambda}[\mathcal{G}]}(x) = \begin{pmatrix} \lambda(g_0, g_0^{-1}g_0)\alpha_{g_0^{-1}g_0} & \lambda(g_0, g_0^{-1}g_1)\alpha_{g_0^{-1}g_1} & \dots & \lambda(g_0, g_0^{-1}g_{n-1})\alpha_{g_0^{-1}g_{n-1}} \\ \lambda(g_1, g_1^{-1}g_0)\alpha_{g_1^{-1}g_0} & \lambda(g_1, g_1^{-1}g_1)\alpha_{g_1^{-1}g_1} & \dots & \lambda(g_1, g_1^{-1}g_{n-1})\alpha_{g_1^{-1}g_{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda(g_{n-1}, g_{n-1}^{-1}g_0)\alpha_{g_{n-1}^{-1}g_0} & \lambda(g_{n-1}, g_{n-1}^{-1}g_1)\alpha_{g_{n-1}^{-1}g_1} & \dots & \lambda(g_{n-1}, g_{n-1}^{-1}g_{n-1})\alpha_{g_{n-1}^{-1}g_{n-1}} \end{pmatrix}. \quad (6)$$

Examples: The truncated polynomial ring $\mathcal{R}_{C_n} = \mathbb{Z}[x]/(x^n - 1)$ is isomorphic to the group ring $\mathbb{Z}C_n$, where $C_n = \langle x : x^n = 1 \rangle$ is the cyclic group of order n. Similarly, the ring $\mathcal{R}_{D_N} = \mathbb{Z}[x, y]/(x^N - 1,\ y^2 - 1,\ xy - yx^{N-1})$ is isomorphic to the group ring $\mathbb{Z}D_N$, where $D_N = \langle x, y \mid x^N = y^2 = 1,\ xy = yx^{N-1} \rangle$ is the dihedral group of order $2N$. In the twisted setting, the ring $\mathcal{R}_{C_n^+} = \mathbb{Z}[x]/(x^n + 1)$ is isomorphic to a twisted group ring $\mathbb{Z}^{\lambda_1}C_n$, and the ring $\mathcal{R}_{D_N^+} = \mathbb{Z}[x, y]/(x^N - 1,\ y^2 + 1,\ xy - yx^{N-1})$ is isomorphic to the twisted group ring $\mathbb{Z}^{\lambda_2}D_N$, where

$$\lambda_1(x^i, x^j) = \begin{cases} -1, & \text{if } i + j \geq n \\ 1, & \text{otherwise} \end{cases}, \quad \lambda_2(y^{j_1}x^{i_1}, y^{j_2}x^{i_2}) = \begin{cases} -1, & \text{for } i_1, i_2 \in \{0, 1, \dots, n-1\} \\ & \text{and } j_1 = j_2 = 1 \\ 1, & \text{otherwise.} \end{cases}$$

Figure 1 illustrates the matrix representation $\tau^{\lambda}(a)$ for an element a from each of these rings, respectively.

$$\tau(a) = \begin{pmatrix} a_0 & a_1 & a_2 & \dots & a_{n-1} \\ a_{n-1} & a_0 & a_1 & \dots & a_{n-2} \\ a_{n-2} & a_{n-1} & a_0 & \dots & a_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & a_3 & \dots & a_0 \end{pmatrix}$$

(a) $\tau(a)$ for $a \in \mathcal{R}_{C_n}$

$$\tau^{\lambda_1}(a) = \begin{pmatrix} a_0 & a_1 & a_2 & \dots & a_{n-1} \\ -a_{n-1} & a_0 & a_1 & \dots & a_{n-2} \\ -a_{n-2} & -a_{n-1} & a_0 & \dots & a_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_1 & -a_2 & -a_3 & \dots & a_0 \end{pmatrix}$$

(b) $\tau^{\lambda_1}(a)$ for $a \in \mathcal{R}_{C_n^+}$

$$\tau(a) = \left(\begin{array}{cccc|cccc} a_0 & a_1 & \dots & a_{N-1} & a_N & a_{N+1} & \dots & a_{2N-1} \\ a_{N-1} & a_0 & \dots & a_{N-2} & a_{N+1} & a_{N+2} & \dots & a_N \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & \dots & a_0 & a_{2N-1} & a_N & \dots & a_{2N-2} \\ \hline a_N & a_{N+1} & \dots & a_{2N-1} & a_0 & a_1 & \dots & a_{N-1} \\ a_{N+1} & a_{N+2} & \dots & a_N & a_{N-1} & a_0 & \dots & a_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{2N-1} & a_N & \dots & a_{2N-2} & a_1 & a_2 & \dots & a_0 \end{array}\right)$$

(c) $\tau(a)$ for $a \in \mathcal{R}_{D_N}$

$$\tau^{\lambda_2}(a) = \left(\begin{array}{cccc|cccc} a_0 & a_1 & \dots & a_{N-1} & a_N & a_{N+1} & \dots & a_{2N-1} \\ a_{N-1} & a_0 & \dots & a_{N-2} & a_{N+1} & a_{N+2} & \dots & a_N \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & \dots & a_0 & a_{2N-1} & a_N & \dots & a_{2N-2} \\ \hline -a_N & -a_{N+1} & \dots & -a_{2N-1} & a_0 & a_1 & \dots & a_{N-1} \\ -a_{N+1} & -a_{N+2} & \dots & -a_N & a_{N-1} & a_0 & \dots & a_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{2N-1} & -a_N & \dots & -a_{2N-2} & a_1 & a_2 & \dots & a_0 \end{array}\right)$$

(d) $\tau^{\lambda_2}(a)$ for $a \in \mathcal{R}_{D_N^+}$

Fig. 1: Matrix representations of a in (twisted) group rings over cyclic and dihedral group.

2.4 NTRU

We start with the conventional definition of NTRU as originally introduced in [22].

Definition 7 (NTRU). *Let n be a prime number and q be a positive integer. Let $f, g \in \mathbb{Z}[x]$ be polynomials of degree at most $n - 1$, sampled from distributions χ_f and χ_g on $\mathcal{R}_{C_n}^q = \mathbb{Z}_q[x]/(x^n - 1)$, respectively, with the condition that*

f is invertible in the ring. Then, $h = f^{-1} * g \pmod q$ represents an NTRU instance. The decisional (n, q, χ_f, χ_g)-NTRU problem asks to distinguish h from a uniformly random element $u \leftarrow_\$ \mathcal{R}^q_{C_n}$. The search (n, q, χ_f, χ_g)-NTRU problem asks to recover (f, g) or a rotated pair $(x^i * f, x^i * g)$, given h.

A matrix variant of the NTRU problem was introduced in [12,19] and is formulated as follows.

Definition 8 (Matrix NTRU). *Let n and q be positive integers. Let $\mathbf{F}, \mathbf{G} \in \mathbb{Z}^{n\times n}$ be matrices sampled from distributions $\chi_{\mathbf{F}}$ and $\chi_{\mathbf{G}}$ on $\mathbb{Z}_q^{n\times n}$, respectively, with the condition that $\mathbf{F}$ is invertible. Then, $\mathbf{H} = \mathbf{F}^{-1}\mathbf{G} \pmod q$ represents a matrix NTRU instance. The decisional $(n, q, \chi_{\mathbf{F}}, \chi_{\mathbf{G}})$-matrix NTRU problem asks to distinguish $\mathbf{H}$ from a uniformly random matrix $\mathbf{U} \leftarrow_\$ \mathbb{Z}_q^{n\times n}$. The search $(n, q, \chi_{\mathbf{F}}, \chi_{\mathbf{G}})$-matrix NTRU problem asks to recover $(\mathbf{F}, \mathbf{G})$, given $\mathbf{H}$.*

Another perspective on NTRU variants is provided by the group ring formulation, known as GR-NTRU [47]. Many variants of NTRU constructed over algebraic rings [22,41,3] can be seen as special cases of GR-NTRU or its twisted versions. For instance, the original NTRU scheme can be interpreted as a GR-NTRU instance where the ring is the ring of integers and the group is a finite cyclic group. This leads to a natural generalization of the NTRU problem over arbitrary algebraic structures, as formalized below.

Definition 9 (GR-NTRU). *Let $\mathcal{G}$ be a finite group of order n, let $\mathcal{R}$ be a ring, and let q be a positive integer. Let $f, g \in \mathcal{R}[\mathcal{G}]$ be two elements sampled from distributions χ_f and χ_g on $(\mathcal{R}/\langle q\rangle)[\mathcal{G}]$, respectively, with the condition that f is invertible in the ring. Then, $h = f^{-1} * g \pmod q$ represents a GR-NTRU instance. The decisional $(\mathcal{G}, \mathcal{R}, q, \chi_f, \chi_g)$-GR-NTRU problem asks to distinguish h from a uniformly random element $u \leftarrow_\$ (\mathcal{R}/\langle q\rangle)[\mathcal{G}]$. The search $(\mathcal{G}, \mathcal{R}, q, \chi_f, \chi_g)$-GR-NTRU problem asks to recover (f, g) or a rotation (with respect to the underlying group structure), given h.*

For any instance of group-ring NTRU, the element f is invertible over $(\mathcal{R}/\langle q\rangle)[\mathcal{G}]$ if and only if the corresponding matrix, defined by the map introduced in Equation (4), is invertible over $\mathcal{R}/\langle q\rangle$ [28, Theorem 2]. This result extends naturally to the twisted GR-NTRU setting, where the matrix defined in Equation (6) can similarly be used to verify the invertibility condition.

Additionally, the private elements f and g are typically sampled with coefficients drawn from a discrete Gaussian distribution with variance $\sigma^2 > 0$. In the original NTRU scheme [22], f and g are sampled as ternary polynomials with approximately $n/3$ coefficients equal to -1, $n/3$ equal to 1, and $n/3$ equal to 0, which can be roughly interpreted as sampling from a discrete Gaussian distribution with $\sigma^2 \approx 2/3$.

Attacks Landscape

– **Search Attacks:** The most straightforward approach is to search for two elements $f', g' \in \mathbb{Z}_q^{\lambda}[\mathcal{G}]$ with small coefficients that satisfy the NTRU equation. Search attacks can be further enhanced using Meet-in-the-Middle techniques [24] and optimized combinatorial methods [39,13].
– **Lattice Attacks:** Instead of exhaustively searching for the private key, one can construct a lattice in which the key lies, and then apply lattice reduction techniques to recover a short vector that may serve as a valid decryption key. Given the public key h, the Coppersmith–Shamir lattice corresponding to the GR-NTRU instance is defined as

$$\mathcal{L}^{CS} = \{(\mathbf{g}, \mathbf{f}) \in \mathbb{Z}^{2n} \mid f * h = g \pmod{q}\}.$$

Since the private key $(\mathbf{g}, \mathbf{f})$ typically has small coefficients, the corresponding vector is expected, with high probability, to be among the shortest in the lattice. Thus, an attacker may construct the lattice $\mathcal{L}^{CS}$ generated by the basis:

$$\mathcal{B}^{CS} = \begin{pmatrix} q\mathbf{I}_n & \mathbf{0}_n \\ \tau^{\lambda}(h) & \mathbf{I}_n \end{pmatrix}, \tag{7}$$

and apply lattice reduction algorithms such as BKZ [43] to extract a short vector that could act as a decryption key. This attack approach is called the **Primal** attack. The cost of lattice reduction primarily depends on the volume of the lattice (q^n) and the ratio of the norm of the private key (g, f) to the expected shortest vector length given by the Gaussian heuristic (Definition 4). This ratio is referred to as the *lattice gap*. A larger lattice gap and smaller lattice volume make lattice reduction attacks more computationally demanding in practice. The primal attack can be combined with search-based attacks, resulting in a **Hybrid** attack. Hybrid attacks are considered the most powerful known methods against NTRU-like schemes employing ternary or sparse keys; consequently, the computational cost of these attacks is a critical factor in parameter selection. For a comprehensive description, we refer the reader to [26].

Lattice-based attacks, including combined strategies such as the hybrid attack, remain the most effective methods against NTRU-like constructions. Table 2 presents a comparison between the cost of the best-known combinatorial attacks and that of lattice-based attacks on various NTRU instances. The column (n, q, w) indicates the number of the coefficients n in the sampled polynomials, the modulus q, and the number of non-zero ternary coefficients w out of n. The column $\mathcal{C}$ indicates the cost of the combinatorial attack without Meet-in-the-Middle (MitM). Odlyzko refers to the MitM cost introduced in [24]. May [Free Mem] and May [Poly Mem] represent the combinatorial attack costs from [39] under free and polynomial memory access assumptions, respectively. Esser et al. indicate the cost of a combinatorial attack under polynomial memory as proposed in [13]. Finally, the Lattice Core-SVP column estimates the lattice attack cost using the classical Core-SVP model $2^{0.292\beta}$, where β is the BKZ blocksize. The table shows that lattice-based techniques are significantly more efficient in practice. Therefore, the selection of parameters in these schemes is influenced by the cost of the lattice attacks, ensuring the desired levels of security.

Table 2: Comparison of the estimated costs of combinatorial and lattice-based attacks against selected NTRU-like schemes from the literature.

(n, q, w)	$\mathcal{C}$	Odlyzko [24]	May [Free Mem] [39]	May [Poly Mem] [39]	Esser et al. [13]	Lattice-Core-SVP
NTRU-HPS [7]						
(509,2048,254)	754	377	227	490	429	**105**
(677,2048,254)	891	445	273	581	492	**144**
(821,4096,510)	1286	643	378	852	750	**178**
NTRU Prime [5]						
(653,4621,288)	925	463	272	599	518	**129**
(761,4591,286)	1003	502	301	653	553	**153**
(857,5167,322)	1131	566	338	735	623	**175**
DiTRU$^+$ [42]						
(2*269,2048,254)	781	391	239	517	453	**111**
(2*347,2048,254)	902	451	280	595	504	**147**
(2*419,4096,510)	1309	654	385	869	766	**182**

3 Our Constructions

This section introduces three constructions: the first is an almost unstructured variant, the second is semi-structured, and the third is fully structured and based on the group ring of the dihedral group.

3.1 Almost Unstructured Variant

We begin by introducing the definitions of the decisional and search problems for the Almost-NTRU (abbreviated as ANTRU) scheme.

Definition 10 (unstructured-ANTRU or u-ANTRU). *Let n be a prime number and q be a positive integer. Let $f \in \mathbb{Z}[x]$ be a polynomial of degree at most $n-1$, sampled from a distribution χ_f, and let $\mathbf{G} \in \mathbb{Z}^{n\times n}$ be sampled from a distribution $\chi_{\mathbf{G}}$. Let f_1 and f_2 be two invertible elements in the ring $\mathcal{R}^q_{C_n^+} = \mathbb{Z}_q[x]/(x^n+1)$, such that $f_1 \leftarrow_{\$} \mathcal{R}^q_{C_n^+}$ and $f_2 = f_1^{-1} * f \pmod q$. Let $\mathbf{F}_1$ and $\mathbf{F}_2$ denote the matrix representations of f_1 and f_2, respectively. Then, the matrix*

$$\mathbf{H} = \mathbf{F}_1^{-1}\mathbf{G}\mathbf{F}_2^{-1} \pmod q \tag{8}$$

is called a u-ANTRU instance.

Definition 11 (Decisional u-ANTRU Problem). *Given the parameters $(n, q, \chi_f, \chi_{\mathbf{G}})$, distinguish $\mathbf{H}$ from a uniformly random matrix $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{n\times n}$.*

Definition 12 (Search u-ANTRU Problem). *Given $\mathbf{H}$, recover a tuple $(f_1, f_2, \mathbf{G})$ such that f_1 and f_2 are invertible that satisfy $f_1 * f_2 = f \pmod q$ and the public key equation (8), for some f drawn from χ_f and invertible, under the same parameter set $(n, q, \chi_f, \chi_{\mathbf{G}})$.*

Consider the public key equation of NTRU (Definition 7): $h = f^{-1} * g \pmod q$. It is easy to observe that f can be written as a product $f = f_1 * f_2$ for some invertible polynomials f_1 and f_2. Substituting this into the public key equation yields $h = f_2^{-1} * f_1^{-1} * g \pmod q$, which can be rearranged as $h = f_1^{-1} * g * f_2^{-1} \pmod q$, since the underlying ring is commutative.

The public key equation of u-ANTRU is similar in form to that of NTRU, but with a crucial distinction: the order of multiplication matters, as the matrix

$\mathbf{G}$ does not commute with the matrices $\mathbf{F}_1$ and $\mathbf{F}_2$. Therefore, while it holds that $\mathbf{H}' = \mathbf{F}_2^{-1}\mathbf{F}_1^{-1}\mathbf{G} = \mathbf{F}_1^{-1}\mathbf{F}_2^{-1}\mathbf{G} \pmod q$, this equivalence does not imply that $\mathbf{H} = \mathbf{H}' \pmod q$.

Nevertheless, one can show that the decisional and search versions of the u-ANTRU problem are at least as hard as the corresponding decisional and search versions of the NTRU problem.

Lemma 1. *Let $\mathcal{A}$ be a PPT (probabilistic polynomial time) algorithm that solves the decisional or search $(n, q, \chi_f, \chi_{\mathbf{G}})$-u-ANTRU problem with advantage ϵ. Then, there exists an algorithm $\mathcal{B}$ that, by invoking $\mathcal{A}$ once with a negligible amount of additional computations, can solve the decisional/search (n, q, χ_f, χ_g)-NTRU problem with the same advantage ϵ.*

Proof. Suppose the algorithm $\mathcal{A}$ can solve the decisional u-ANTRU problem with advantage ϵ. Then, trivially, an NTRU instance $h = f^{-1} * g \pmod q$ can be interpreted as an u-ANTRU instance with $f_1 = f$, $\mathbf{G}$ as the matrix representation of $g \in \mathcal{R}^q_{C_n^+}$, and $f_2 = 1$. Therefore, if the algorithm $\mathcal{A}$ can solve the decisional u-ANTRU problem with advantage ϵ, it should also be able to distinguish an NTRU instance with the same advantage. However, the oracle $\mathcal{A}$ may detect that the queried instance always falls under this special case. To address this, we randomize the input as follows.

Let h be an NTRU instance computed as $h = f^{-1} * g \pmod q$. Then, algorithm $\mathcal{B}$ proceeds as follows:

- Convert h into matrix form: compute $\mathbf{H} = \mathbf{F}^{-1}\mathbf{G} \pmod q$, as described in Figure 1.
- Generate an elementary matrix $\mathbf{I}_{e_n}$ by permuting the rows of the identity matrix $\mathbf{I}_n$, and compute $\mathbf{H}' = \mathbf{H}\mathbf{I}_{e_n} = \mathbf{F}^{-1}(\mathbf{G}\mathbf{I}_{e_n}) \pmod q$. Let $\mathbf{G}' = \mathbf{G}\mathbf{I}_{e_n}$.
- Sample a random invertible polynomial $f' \leftarrow_{\$} \mathcal{R}^q_{C_n^+}$, let $\mathbf{F}'$ be its matrix representation, and compute

$$\mathbf{H}'' = \mathbf{F}'\mathbf{H}'\mathbf{F}'^{-1} = (\mathbf{F}\mathbf{F}'^{-1})^{-1}\mathbf{G}'\mathbf{F}'^{-1} \pmod q.$$

- Query algorithm $\mathcal{A}$ once with the randomized instance $\mathbf{H}''$ representing an u-ANTRU sample.
- Algorithm $\mathcal{B}$ returns 1 if algorithm $\mathcal{A}$ outputs 1 (indicating that h is an NTRU instance), and returns 0 otherwise.

For the search problem, algorithm $\mathcal{B}$ follows the same steps to generate $\mathbf{H}''$ and provides it to algorithm $\mathcal{A}$, which solves the search problem of u-ANTRU. If $\mathcal{A}$ returns a solution tuple $(f_1, f_2, \mathbf{G}') = (f * f'^{-1}, f', \mathbf{G}\mathbf{I}_{e_n})$, then algorithm $\mathcal{B}$ recovers $f = f_1 * f_2 \pmod q$ and $\mathbf{G} = \mathbf{G}'\mathbf{I}_{e_n}^{-1}$. Since g corresponds to the first row of $\mathbf{G}$, algorithm $\mathcal{B}$ finally returns (f, g) as the solution to the NTRU search problem. □

Encryption Scheme The encryption scheme based on u-ANTRU is similar to the general encryption scheme employed in NTRU-HPS, and is outlined in Figure 2. The notations used have the following meanings:

- $\mathcal{T}$ denotes the set of ternary elements in $\mathcal{R}_{C_n^+}$.
- $\mathcal{T}(a, b)$ denotes the set of elements in $\mathcal{R}_{C_n^+}$ with a coefficients equal to 1, b coefficients equal to -1, and the remaining coefficients set to 0.
- $\mathcal{T}_{\text{matrix}}(a, b)$ denotes the set of matrices in $\mathbb{Z}^{n\times n}$ such that each column has a entries equal to 1, b entries equal to -1, and the rest equal to 0.
- $\mathcal{L}_f = \mathcal{T}(d_f + 1, d_f), \mathcal{L}_{\mathbf{G}} = \mathcal{T}_{\text{matrix}}(d_g, d_g), \mathcal{L}_r = \mathcal{T}(d_r, d_r)$, and $\mathcal{L}_m = \mathcal{T}$.
- Sampler instantiated with a set and a seed deterministically returns an element from the underlying set associated with the seed.
- Centerlift$(\cdot, q)$ returns an element whose lie in the range $(-q/2, q/2]$.

KeyGen(1^λ)

1: initialize Sampler with $\mathcal{L}_f$ and seed_f
2: do $f \leftarrow$ Sampler until f is invertible in $\mathcal{R}^q_{C_n^+}$
3: initialize Sampler with $\mathcal{R}^q_{C_n^+}$ and seed_{f_1}
4: do $f_1 \leftarrow$ Sampler until f_1 is invertible in $\mathcal{R}^q_{C_n^+}$
5: let $f_2 = f_1^{-1} * f \pmod q$.
6: let f_p denotes the inverse of f in $\mathcal{R}^p_{C_n^+}$
7: let $\mathbf{F}_1$ and $\mathbf{F}_2$ denote the matrix representations of f_1 and f_2
8: initialize Sampler with $\mathcal{L}_{\mathbf{G}}$ and $\mathsf{seed}_{\mathbf{G}}$
9: $\mathbf{G} \leftarrow$ Sampler
10: $\mathbf{H} = \mathbf{F}_1^{-1}\mathbf{G}\mathbf{F}_2^{-1} \pmod q$
11: **return** $((f_1, f_2, f_p), \mathbf{H})$

Encrypt$(\mathbf{H}, m, \mathsf{seed}_{r_1}, \mathsf{seed}_{r_2})$

1: initialize Sampler with $\mathcal{L}_r$ and seed_{r_1}
2: $r_1 \leftarrow$ Sampler
3: initialize Sampler with $\mathcal{L}_r$ and seed_{r_2}
4: $r_2 \leftarrow$ Sampler
5: let $\mathbf{R}_1$ and $\mathbf{R}_2$ denote the matrix representations of r_1 and r_2
6: let $\mathbf{M}$ indicates the matrix representation of m
7: $\mathbf{C} = p\mathbf{R}_1\mathbf{H}\mathbf{R}_2 + \mathbf{M} \pmod q$
8: **return** $\mathbf{C}$

Decrypt$((f_1, f_2, f_p), \mathbf{C})$

1: let $\mathbf{F}_1$ and $\mathbf{F}_2$ denote the matrix representations of f_1 and f_2
2: let $\mathbf{F}_p$ denotes the matrix representation of f_p
3: $\mathbf{T} = \mathbf{F}_1\mathbf{C}\mathbf{F}_2 \pmod q$
4: $\mathbf{A} =$ Centerlift$(\mathbf{T}, q)$
5: $\mathbf{M}' = \mathbf{F}_p\mathbf{A} \pmod p$
6: let m' equals the first row of $\mathbf{M}'$
7: $m \leftarrow$ Centerlift(m', p)
8: **return** m

Fig. 2: A sketched description of the encryption scheme of u-ANTRU.

3.2 Attack Strategy

Lower Bound: Since the public key equation of u-ANTRU is inherently supported by the lower-bound security of the standard NTRU assumption, the best an attacker can hope for is to transform the u-ANTRU equation $\mathbf{H} = \mathbf{F}_1^{-1}\mathbf{G}\mathbf{F}_2^{-1} \pmod q$ into a commutative analog of the form $\mathbf{H}' = \mathbf{F}^{-1}\mathbf{G}' \pmod q$, where $\mathbf{F} = \mathbf{F}_1\mathbf{F}_2 \pmod q$ and $\mathbf{G}'$ is a matrix with small entries. In such a case, the attacker could attempt lattice reduction techniques on $\mathbf{H}'$ to recover $\mathbf{F}$, which might subsequently aid in message recovery. Therefore, when considering both the key and ciphertext equations, the attacker's task reduces to constructing a mapping that transforms the noncommutative equation into a commutative

one. Let $\mathbf{RC}^{n\times n}$ denote the set of all right circulant matrices over $\mathbb{Z}^{n\times n}$. The attacker would then be seeking a map $\theta : \mathbb{Z}^{n\times n} \to \mathbf{RC}^{n\times n}$ satisfying the following properties:

- It is not the zero map.
- It is linear (a requirement for enabling decryption).
- It is both left- and right-linear with respect to elements belonging to the commutative subring (in this case, $\mathbf{RC}^{n\times n}$).
- It maps a matrix $\mathbf{G}$ with small entries to a matrix $\mathbf{G}'$ whose entries are bounded by $\alpha g_{\max}$, where α is a small integer and $g_{\max} = \max_{i,j} |\mathbf{G}^{(i,j)}|$, ensuring applicability of lattice reduction.
- It is easily (polynomial time) invertible on the images of elements representing the message space.

Upon identifying such a map θ, the attacker can recover $\mathbf{F}$ by applying lattice reduction to the lattice generated by $\begin{pmatrix} q\mathbf{I}_n & \mathbf{0}_n \\ \theta(\mathbf{H}) & \mathbf{I}_n \end{pmatrix}$. Subsequently, the attacker can compute

$$\begin{aligned} \mathbf{F}\theta(\mathbf{C}) &= p\mathbf{R}_1\mathbf{R}_2\mathbf{F}\theta(\mathbf{H}) + \mathbf{F}\theta(\mathbf{M}) \pmod q \\ &= p\mathbf{R}_1\mathbf{R}_2\theta(\mathbf{G}) + \mathbf{F}\theta(\mathbf{M}) \pmod q, \end{aligned}$$

which, when reduced modulo p, yields $\mathbf{F}\theta(\mathbf{M})$. Since $\mathbf{F}$ is invertible and by definition, θ is invertible on $\mathbf{M}$, the attacker can then reconstruct $\mathbf{M}$. In the case of u-ANTRU, a possible construction of such a map θ that satisfies the required properties is given by

$$\theta(\mathbf{X}) = \sum_{i=0}^{k} \mathbf{A}_i\mathbf{X}\mathbf{B}_i, \qquad \mathbf{A}_i, \mathbf{B}_i \in \mathbf{RC}^{n\times n}.$$

However, the task of identifying such a map θ (if it exists) remains highly nontrivial and is strongly determined by the algebraic structure of the underlying ring. Consequently, we next present a general attack strategy that applies in situations where no such map can be efficiently determined.

Key Recovery Attack The combinatorial attack against the secret key involves first guessing a ternary polynomial $f \in \mathcal{L}_f$, followed by guessing the first factor $f_1 \in \mathcal{L}_f$. Once f_1 is determined, the second factor f_2 can be computed directly. For the correct guess, we obtain a matrix

$$\mathbf{G} = \mathbf{F}_1\mathbf{H}\mathbf{F}_2 \pmod q,$$

whose entries are small. The overall cost of such a combinatorial attack on the key is prohibitively large. For instance, when $\mathcal{L}_f = \mathcal{T}(d_f + 1, d_f)$, the search complexity is given by

$$q^n \binom{n}{d_f + 1} \binom{n - d_f - 1}{d_f}.$$

Message Recovery Attack Attacking the scheme purely via lattice reduction is computationally expensive. The ciphertext equation is given by

$$\mathbf{C} = p\mathbf{R}_1\mathbf{H}\mathbf{R}_2 + \mathbf{M} \pmod q,$$

where

$$\mathbf{R}_1 = \begin{pmatrix} r_{10} & r_{11} & \dots & r_{1(n-1)} \\ -r_{1(n-1)} & r_{10} & \dots & r_{1(n-2)} \\ \vdots & \vdots & \ddots & \vdots \\ -r_{11} & -r_{12} & \dots & r_{10} \end{pmatrix}, \quad \mathbf{R}_2 = \begin{pmatrix} r_{20} & r_{21} & \dots & r_{2(n-1)} \\ -r_{2(n-1)} & r_{20} & \dots & r_{2(n-2)} \\ \vdots & \vdots & \ddots & \vdots \\ -r_{21} & -r_{22} & \dots & r_{20} \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} h_0 & h_1 & \dots & h_{n-1} \\ h_n & h_{n+1} & \dots & h_{2n-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n(n-1)} & h_{n(n-1)+1} & \dots & h_{n^2-1} \end{pmatrix}.$$

This matrix equation can be equivalently represented in polynomial form (see Appendix F) as

$$\overrightarrow{\mathbf{C}}^{\text{poly}} = \underbrace{\begin{pmatrix} h_0(x) & h_1(x) & \dots & h_{n-1}(x) \\ h_1(x) & h_2(x) & \dots & -h_0(x) \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-1}(x) & -h_0(x) & \dots & -h_{n-2}(x) \end{pmatrix}}_{\mathbf{H}^{\text{poly}}} \begin{pmatrix} r_{10}\, r_2(x) \\ r_{11}\, r_2(x) \\ \vdots \\ r_{1(n-1)}\, r_2(x) \end{pmatrix} + \begin{pmatrix} m(x) \\ x * m(x) \\ \vdots \\ x^{n-1} * m(x) \end{pmatrix}, \quad (9)$$

where $h_i(x)$ denotes the polynomial corresponding to the coefficient vector of the i^{th} column of the matrix $\mathbf{H}$, and $m(x)$ is the polynomial obtained from the coefficient vector of the first row of the matrix $\mathbf{M}$. Naively, the message can be recovered by solving the shortest vector problem (SVP) in a $(2n^2+1)$-dimensional lattice generated by $\begin{pmatrix} q\mathbf{I}_{n^2} & \mathbf{0}_{n^2} & 0_{n^2} \\ \boldsymbol{\mathcal{H}}^{\text{poly}} & \mathbf{I}_{n^2} & 0_{n^2} \\ \overrightarrow{\mathbf{C}}^{Tr} & \mathbf{0}_{n^2} & 1 \end{pmatrix}$, where $\boldsymbol{\mathcal{H}}^{\text{poly}}$ is the integer matrix representation corresponding to the polynomial matrix $\mathbf{H}^{\text{poly}}$, and $\overrightarrow{\mathbf{C}}$ denotes the vectorization of the ciphertext matrix $\mathbf{C}$. In the commutative case, the lattice dimension would reduce to $(2n+1)$. However, message recovery in the noncommutative setting is essentially equivalent to solving an over-structured Module-LWE instance. An attacker may therefore attempt to construct a homomorphism that reduces the lattice dimension from $2n^2+1$ to a smaller dimension, thereby extracting useful information about the message from lower-dimensional lattices. Nevertheless, any such homomorphism (if it exists) must map short vectors in the original lattice to short vectors in the reduced lattice, which implies that the dimension cannot be reduced by a large factor.

Another observation is that the error polynomials are cyclic rotations of one another, a structure that may enable the construction of multiple noiseless equations, potentially leading to more efficient message-recovery attacks. In order to avoid such attacks, we modify the general design sketched in Figure 2 so that both f_1 and f_2 are sampled as invertible elements from $\mathcal{R}^q_{C_n^+}$, specifically from the space $\mathcal{L}_f = \mathcal{T}(d_f+1, d_f)$, and $\mathbf{M}$ is sampled as a ternary matrix similar to $\mathbf{G}$, rather than as a cyclic matrix. In this setting, the product $f_1 * f_2$ does not yield a ternary polynomial but instead a polynomial with larger coefficients. Consequently, during decryption, the value of q must be larger compared to the variant presented in Definition 10. Nevertheless, the u-ANTRU problem is still

regarded as a two-level NTRU problem, and the u-ANTRU search problem is modified accordingly.

Definition 13 (Refined u-ANTRU instance). *Let n be a prime, q a positive integer, and $f_1, f_2 \in \mathbb{Z}[x]$ (with $\deg(f_i) \leq n-1$) drawn from χ_f, and let $\mathbf{G} \in \mathbb{Z}^{n\times n}$ be drawn from $\chi_{\mathbf{G}}$. Let $\mathbf{F}_1$ and $\mathbf{F}_2$ denote the matrix representations of f_1 and f_2, respectively; then $\mathbf{H} = \mathbf{F}_1^{-1}\mathbf{G}\mathbf{F}_2^{-1} \pmod q$ is called an u-ANTRU instance.*

Definition 14 (Refined Search u-ANTRU Problem). *Given $\mathbf{H}$, a modified u-ANTRU instance, recover $(f_1, f_2, \mathbf{G})$ such that f_1, f_2 are invertible and drawn from χ_f.*

For the modified construction of u-ANTRU, the attacker can attempt the following strategies:

(i) **Key Attack:** guess f_1. For the correct guess, $\mathbf{F}_1^{-1}\mathbf{H} \pmod q$ yields an NTRU instance, which can be recognized using the best available lattice reduction techniques.
(ii) **Message Attack:** guess r_1. For the correct guess, the pair $(\mathbf{R}_1\mathbf{H}, \mathbf{C})$ forms an LWE instance, which can likewise be identified via the best lattice reduction techniques.

Hence, the most effective strategy combines combinatorial guessing with lattice-based attacks. It is worth noting that improved search attacks, such as the meet-in-the-middle approach, cannot be applied directly since $\mathbf{F}_1\mathbf{H} \pmod q$ is not a matrix with small coefficients.

3.3 Semi-structured ANTRU or semi-ANTRU

The semi-structured construction balances the memory and time requirements, as well as the structural constraints, compared to the almost-unstructured variant introduced above in Subsection 3.1 and the fully structured variant introduced in the next Subsection 3.4. Rather than sampling $\mathbf{G}$ with a random structure, the semi-structured variant samples g (whose corresponding matrix form is $\mathbf{G}$) from the group ring constructed over the twisted dihedral group[3]:

$$\mathcal{R}^q_{D_N^+} \;=\; \mathbb{Z}_q D_N^+ \;\approx\; \frac{\mathbb{Z}_q[x,y]}{(x^N-1,\; y^2+1,\; xy-yx^{N-1})}. \tag{10}$$

Therefore, the key generation, encryption, and decryption of semi-ANTRU follow the same steps as sketched in Figure 2, with the note that $f_1, f_2, r_1, r_2 \in \mathcal{R}_{C_{2N}^+}$ and $g \in \mathcal{R}_{D_N^+}$. One can observe that the matrix representations of elements in $\mathcal{R}_{C_{2N}^+}$ and $\mathcal{R}_{D_N^+}$ share a common structure of the form $\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ -\mathbf{B} & \mathbf{A} \end{pmatrix}$. In the case of

[3] We use the rings $\mathcal{R}^q_{C_n^+}$ and $\mathcal{R}^q_{D_N^+}$ rather than $\mathcal{R}^q_{C_n}$ and $\mathcal{R}^q_{D_N}$, as the structure associated with these rings avoids unnecessary homomorphisms that could make solving the SVP problem easier in lattices of smaller dimensions. See Appendix G.

$\mathcal{R}_{C_{2N}^+}$, the entire matrix is (twisted) right circulant. For $\mathcal{R}_{D_N^+}$, the submatrix $\mathbf{A}$ is right circulant, while $\mathbf{B}$ is left circulant (see Figure 1). Moreover, the product $\mathbf{H} = \mathbf{F}_1^{-1}\mathbf{G}\mathbf{F}_2^{-1}$ and as a result $\mathbf{C} = p\mathbf{R}_1\mathbf{H}\mathbf{R}_2 + \mathbf{M}$, where $\mathbf{M} = \begin{pmatrix} \mathbf{M}_0 & \mathbf{M}_1 \\ -\mathbf{M}_1 & \mathbf{M}_0 \end{pmatrix}$ is a random ternary matrix, exhibits a similar block pattern, although the resulting submatrices $\mathbf{A}$ and $\mathbf{B}$ may not retain any clearly identifiable structure. Consequently, the public key and the ciphertext are compressed for the semi-structured variant compared to the almost-unstructured construction, and one can further optimize the matrix multiplications due to their similar block structure.

3.4 Fully Structured ANTRU or s-NTRU

The fully structured construction is inspired by the first unsuccessful attempt to build a noncommutative variant of NTRU introduced by Hoeffstein and Silverman [23](refer to Appendix D). For a dihedral group of order $2N$ defined as $D_N = \langle x, y \mid x^N = 1, y^2 = 1,\ xy = yx^{N-1} \rangle$, let

$$\mathcal{R}_{D_N^+} = \mathbb{Z}D_N^+ \approx \frac{\mathbb{Z}[x,y]}{(x^N - 1, y^2 + 1, xy - yx^{N-1})},\ \mathcal{R}_{D_N^+}^q = \mathbb{Z}_q D_N^+,\ \mathcal{R}_{D_N^+}^p = \mathbb{Z}_p D_N^+. \quad (11)$$

We will consider N to be an odd prime for the rest of this work. An element $f \in \mathcal{R}_{D_N^+}$ can be written as $a_0(x) + ya_1(x)$, where $a_0(x), a_1(x) \in \mathcal{R}_{C_N} = \frac{\mathbb{Z}[x]}{(x^N-1)}$. We denote $\overline{a_0(x)} = a_0(x^{N-1})$ as the conjugate of $a_0(x)$. In other words, if $\mathbf{a}_0 = (a_{00}, a_{01}, \ldots, a_{0N-1})$ is the vector form of $a_0(x)$, then $\overline{\mathbf{a}_0} = (a_{00}, a_{0N-1}, \ldots, a_{01})$ is the vector form of $\overline{a_0(x)}$. Let $f = a_0(x) + ya_1(x)$ and $g = b_0(x) + yb_1(x)$ be two elements from $\mathcal{R}_{D_N^+}$. Then, the operations are defined as:

- **Addition:** $f + g = \big(a_0(x) + b_0(x)\big) + y\big(a_1(x) + b_1(x)\big)$.
- **Multiplication:** $f * g = a_0(x) * b_0(x) - \overline{a_1(x)} * b_1(x) + y\big(a_1(x) * b_0(x) + \overline{a_0(x)} * b_1(x)\big)$.

Additionally, the construction of s-ANTRU requires a commutative subring for the decryption process to work. Let $\mathcal{S}$ indicates the commutative subring of $\mathcal{R}_{D_N^+}$ defined as

$$\mathcal{S} = \{f \in \mathcal{R}_{D_N^+} \mid fy = yf\}. \quad (12)$$

In other words, $\mathcal{S}$ contains all elements $f = a_0(x) + ya_1(x) \in \mathcal{R}_{D_N^+}$ such that $a_i(x) = a_i(x^{N-1})$ for $i = 0, 1$. Therefore, elements in $\mathcal{S}$ have the form

- **Symmetric:** $f = a_{00} + ya_{10} + \sum_{i=1}^{(N-1)/2}(a_{0i} + ya_{1i})(x^i + x^{N-i})$,

and are referred to as symmetric elements. We further define a subset $\mathcal{S}^-$ of $\mathcal{R}_{D_N^+}$ as

$$\mathcal{S}^- = \{f \in \mathcal{R}_{D_N^+} \mid fy = -yf\}, \quad (13)$$

i.e., all elements $f = a_0(x) + ya_1(x) \in \mathcal{R}_{D_N^+}$ such that $a_i(x) = -a_i(x^{N-1})$ for $i = 0, 1$, and thus have the form

– **Antisymmetric:** $f = \sum_{i=1}^{(N-1)/2}(a_{0i} + ya_{1i})(x^i - x^{N-i})$.

Observe that the set $\mathcal{S}^-$ is not a subring; however, it is closed under addition. The elements in $\mathcal{S}^-$ are referred to as antisymmetric. The following are the properties of symmetric and antisymmetric elements:

– For any element $a \in \mathcal{R}_{D_N^+}$, we have, $a - yay \in \mathcal{S}$ and $a + yay \in \mathcal{S}^-$.
– For $a \in \mathcal{S}$, we have, $a - yay = 2a$ and $a + yay = 0$.
– For $a \in \mathcal{S}^-$, we have, $a - yay = 0$ and $a + yay = 2a$.
– For $a \in \mathcal{S},\ b \in \mathcal{S}^-$, we have, $a * b, b * a \in \mathcal{S}^-$ and $a * b \neq b * a$ (in general).

It is easy to verify that the cost of schoolbook multiplication of two elements f and g in $\mathcal{R}_{D_N^+}$ is $(N+1)^2$ when both f and g belong to $\mathcal{S}$; $(N-1)^2$ when both f and g belong to $\mathcal{S}^-$; and N^2-1 when f is in $\mathcal{S}$ and g is in $\mathcal{S}^-$, or vice versa. See Appendix E for the general expressions for multiplication of structured elements from $\mathcal{S}$ or $\mathcal{S}^-$ in the dihedral group ring.

The inverse of the element $f = a_0(x) + ya_1(x)$ is defined as $f^{-1} = \tilde{a}_0(x) + y\tilde{a}_1(x)$, satisfying $f * f^{-1} = f^{-1} * f = 1$. An efficient inversion algorithm for the ring $\mathcal{R}_{D_N^+}$ is provided in [42, Algorithm 1] that is a direct adaptation of [41, Algorithm 1]. For the sake of completeness, we reiterate the required inversion algorithm in Algorithm 1 that states an element $f = a_0(x) + ya_1(x)$ is invertible in $\mathcal{R}_{D_N^+}$ if and only if

$$\det(f) = a_0(x) * \overline{a_0(x)} + a_1(x) * \overline{a_1(x)} \pmod{x^N - 1}$$

is invertible over $\mathcal{R}_{C_N}$. We refer the reader to [41, Theorem 2] for the proof. Further, the inverse of $f \in \mathcal{S}$ is an element that belongs to the same commutative

Algorithm 1 Inversion over $\mathcal{R}_{D_N^+}$

Input: $f = a_0(x) + ya_1(x) \in \mathcal{R}_{D_N^+}$
Output: $f^{-1} = \tilde{a}_0(x) + y\tilde{a}_1(x) \in \mathcal{R}_{D_N^+}$ or a failure
$t_1(x) \leftarrow a_0(x) * \overline{a_0(x)}$
$t_2(x) \leftarrow a_1(x) * \overline{a_1(x)}$
$\det(f) \leftarrow t_1(x) + t_2(x)$ ▷ Coefficientwise addition in $\mathcal{R}_{D_N^+}$
$t_3(x), found \leftarrow \mathsf{inverseRC_N}(\det(f))$ ▷ Inversion over $\mathcal{R}_{C_N}$
if not found **then**
    return failure
end if
$\tilde{a}_0(x) \leftarrow \overline{a_0(x)} * t_3(x)$
$\tilde{a_1}(x) \leftarrow -a_1(x) * t_3(x)$
return $f^{-1} = \tilde{a}_0(x) + y\tilde{a}_1(x)$

subring $\mathcal{S}$ as shown in Lemma 2.

Lemma 2. *Let $f = a_0(x) + ya_1(x) \in \mathcal{S}$. Then, its inverse is of the form $f^{-1} = \tilde{a}_0(x) + y\tilde{a}_1(x) \in \mathcal{S}$.*

Proof. Since $f \in \mathcal{S}$, it satisfies $\overline{a_0(x)} = a_0(x)$ and $\overline{a_1(x)} = a_1(x)$. As f^{-1} is the inverse of f, we have $f * f^{-1} = f^{-1} * f = 1$, which implies

$$a_0(x) * \tilde{a}_0(x) - a_1(x) * \tilde{a}_1(x) = a_0(x) * \tilde{a}_0(x) - \overline{\tilde{a}_1(x)} * a_1(x) = 1,$$
$$a_1(x) * \tilde{a}_0(x) + a_0(x) * \tilde{a}_1(x) = \tilde{a}_1(x) * a_0(x) + \overline{\tilde{a}_0(x)} * a_1(x) = 0.$$

From the above equalities, it follows that $\tilde{a}_0(x) = \overline{\tilde{a}_0(x)}$ and $\tilde{a}_1(x) = \overline{\tilde{a}_1(x)}$, which implies that $f^{-1} \in \mathcal{S}$. □

Design of s-ANTRU The s-ANTRU framework (Figure 3), though inspired by Silverman and Hoffstein's noncommutative variant [23], diverges significantly in its selection of key and message spaces and encryption–decryption mechanisms. Preserving the core NTRU design principles and simultaneously leveraging noncommutativity, s-ANTRU mitigates direct lattice attacks and eliminates vulnerabilities in the Silverman–Hoffstein construction, notably those exploited by Coppersmith [9], through a systematic redefinition of the underlying algebraic structures. For clarity, let us redefine and simplify some notations.

- Let $\mathcal{R} = \mathcal{R}_{D_N^+}$, $\mathcal{R}_p = \mathcal{R}^p_{D_N^+}$, and $\mathcal{R}_q = \mathcal{R}^q_{D_N^+}$, where $p = 3$ is fixed.
- Define the following sets used for sampling: $\mathcal{L}_f = \mathcal{T}(d_f + 1, d_f) \cap \mathcal{S}$, $\mathcal{L}_g = \mathcal{T}(d_g, d_g) \cap \mathcal{S}^-$, $\mathcal{L}_r = \mathcal{T}(d_r, d_r) \cap \mathcal{S}$, and $\mathcal{L}_m = \mathcal{T}(d_m, d_m) \cap \mathcal{S}^-$.
- The definitions of $\mathcal{T}$, Sampler, and Centerlift from Section 3.1 are extended to elements of the dihedral group ring $\mathcal{R}$.

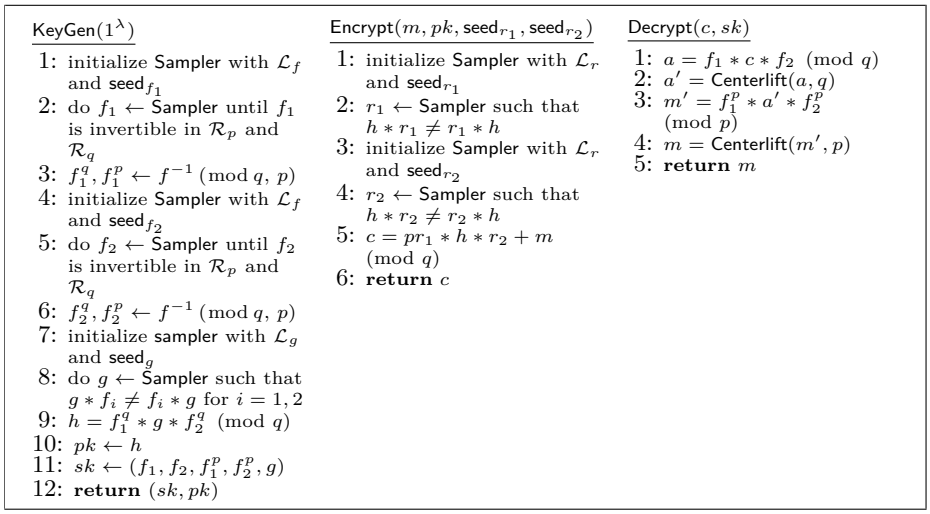

Fig. 3: A sketched description of the s-ANTRU scheme.

4 Security Analysis

Security Against Coppersmith's Attack The primary attack on Hoffstein and Silverman's scheme stems from the weakness that an adversary can construct a suitable map that converts the noncommutative public key equation, $h = f_1 * g * f_2 \pmod{q}$, into a commutative one, and the fact that $f_1 = f_2^{-1} \pmod{q}$ directly reveals the secret g. Specifically, given the public key h, compute

$$\begin{aligned}\theta(h) = h + yhy = pf_1 * (g + ygy) * f_1^{-1} \,(\mathrm{mod} q) &= pf_1 * f_1^{-1} * (g + ygy) \,(\mathrm{mod} q) \\ &= p(g + ygy) \,(\mathrm{mod} q),\end{aligned}$$

where the third equality holds since $f_1 \in \mathcal{S}$ and $g + ygy \in \mathcal{S}$ for $g \in \mathcal{R}_{D_N}$. This partial leakage $g + ygy$ can be exploited to recover g. Refer to Appendix D for more details on this attack.

In contrast, for s-ANTRU, these natural maps that previously enabled the reduction to a commutative case yield only trivial outcomes:

- The map $\theta : \mathcal{R} \to \mathcal{S}^-$, defined as $\theta(a) = a + yay$, gives

$$\theta(h) = h + yhy = f_1^{-1} * (g + ygy) * f_2^{-1} \quad = f_1^{-1} * (2g) * f_2^{-1} = 2h \pmod q,$$

  since $g + ygy = 2g \in \mathcal{S}^-$ does not commute with f_1^{-1} or f_2^{-1}. Thus, only a scaled version of h is obtained.
- The map $\theta' : \mathcal{R} \to \mathcal{S}$, defined as $\theta'(a) = a - yay$, gives

$$\theta'(h) = h - yhy = f_1^{-1} * (g - ygy) * f_2^{-1} = (f_1 * f_2)^{-1} * (g - ygy) \pmod q.$$

  Therefore, the resulting equation is a GR-NTRU equation. As a result, the adversary must solve an NTRU-like problem and, consequently, a hard lattice problem to recover partial information about the secrets. However, our choice of sampling $g \in \mathcal{S}^- \subset \mathsf{Kernel}(\theta')$ (the set of elements in $\mathcal{R}$ mapped to zero under θ') ensures that $g - ygy = 0$. Consequently, $\theta'(h) = 0$, yielding no useful information. This eliminates any possibility of exploiting this map to perform lattice attacks against the public key h or to distinguish it from a random element of the subset $\mathcal{S}^-$.

The encryption and decryption procedures of s-ANTRU are more closely aligned with those of NTRU and differ fundamentally from [23], where the ciphertext consists of a pair of elements in $\mathcal{R}^q_{D_N}$. Although lattice attacks are not directly applicable to Hoffstein and Silverman's message encryption, Coppersmith's algebraic key attack enables the successful recovery of the message from the ciphertext. In contrast, s-ANTRU prevents Coppersmith's key attack, and thus no analogous message-recovery strategy can be applied. Furthermore, the aforementioned maps also fail to yield any useful information, since

$$\theta(c) = 2c \quad \text{and} \quad \theta'(c) = 0, \text{ as } c \in \mathcal{S}^-.$$

Security Against Lattice Attacks More generally, the goal of an adversary is to construct a map $\theta : \mathcal{R} \to \mathcal{R}$ that reduces the equation $h = f_1^{-1} * g * f_2^{-1}$ into an NTRU-like equation $\theta(h) = f'^{-1} * g' \pmod q$ such that f' and g' have small coefficients, in order to perform lattice attacks. Given the fact that f_1 and f_2 are small symmetric elements, the most plausible approach is to keep them intact and attempt to map g to some small symmetric element $\theta(g)$ that commutes with f_i, thereby inducing an NTRU-like equation $\theta(h) = (f_1 * f_2)^{-1} * \theta(g) \pmod q$, as also demonstrated in the maps discussed above. It should be noted that the cost of lattice reduction in this case is slightly higher compared to the typical GR-NTRU, since

$$\|(f_1 * f_2, \theta(g))\| > \|(f, g)\|$$

for random ternary f and g. Concretely, the adversary seeks $\alpha_i, \beta_i \in \mathcal{R}$, for $i = 1, 2, \ldots, k$, such that

$$\theta(h) = \sum_{i=1}^{k} \alpha_i * h * \beta_i = \sum_{i=1}^{k} \alpha_i * (f_1^{-1} * g * f_2^{-1}) * \beta_i = f_1^{-1} * \theta(g) * f_2^{-1}$$

holds, where $\theta(g) \in \mathcal{S}$ is small. For the second equality to hold, α_i and β_i must commute with f_1 and f_2. Since the f_i are sampled uniformly at random from $\mathcal{S}$ and remain secret, the only feasible choice is $\alpha_i, \beta_i \in \mathcal{S}$ in order to commute with f_i so that

$$\sum_{i=1}^{k} \alpha_i * (f_1^{-1} * g * f_2^{-1}) * \beta_i = f_1^{-1} * \left(\sum_{i=1}^{k} \alpha_i * g * \beta_i \right) * f_2^{-1} = f_1^{-1} * \theta(g) * f_2^{-1}.$$

However, in that case, $\theta(g) \in \mathcal{S}^-$, since $\alpha_i, \beta_i \in \mathcal{S}$ and $g \in \mathcal{S}^-$. While simultaneously requiring $\theta(g) \in \mathcal{S}$ forces $\theta(g) = 0$. Therefore, constructing a map that reduces the s-ANTRU public key equation to an NTRU-like form appears computationally intractable. Nonetheless, whether alternative mappings might circumvent this obstacle remains an open research question.

Moreover, a similar technique as discussed in Section 2 can be applied to convert the problem to solving SVP in an $(8N^2 + 1)$-dimensional lattice generated by the matrix $\begin{pmatrix} q\mathbf{I}_{4N^2} & \mathbf{0}_{4N^2} & 0_{4N^2} \\ \boldsymbol{\mathcal{H}}^{\text{poly}} & \mathbf{I}_{4N^2} & 0_{4N^2} \\ \overrightarrow{\mathbf{C}}^{Tr} & \mathbf{0}_{4N^2} & 1 \end{pmatrix}$, where $\boldsymbol{\mathcal{H}}^{\text{poly}}$ is the integer matrix representation of the polynomial matrix $\mathbf{H}^{\text{poly}}$ defined in equation (43). Exploiting the algebraic structure of symmetric and antisymmetric elements, the dimension of the lattice can be reduced by approximately half. However, for the parameter sets specified in Section 5, performing lattice reduction in such high-dimensional spaces remains computationally prohibitive and incurs a cost exceeding that of other possible attacks. Consequently, under current knowledge, lattice-based attacks do not constitute the most effective threat to s-ANTRU, barring any future advances that overcome the reduction barriers.

Hardness Assumptions and Security Proof The design modifications introduced by the noncommutative setting of the twisted dihedral group ring necessitate adapting the standard Ring-LWE and NTRU assumptions to this new framework. This subsection formally defines the *decisional s-ANTRU* and *search s-RLWE* assumptions and presents a reduction showing that breaking the OW-CPA security of the proposed scheme is at least as hard as solving these problems.

Definition 15 (s-ANTRU distribution). *Let $\mathcal{R}_q = \mathcal{R}^q_{D_N^+}$, χ_f be a distribution on $\mathcal{S}$, and χ_g be a distribution on $\mathcal{S}^-$. The structured-ANTRU or s-ANTRU distribution is defined as*

$$s - ANTRU_{\mathcal{R}_q, \chi_f, \chi_g} = \left\{ h = f_1^{-1} * g * f_2^{-1} \mid f_1, f_2 \leftarrow \chi_f, g \leftarrow \chi_g \right\}. \tag{14}$$

Definition 16 (s-ANTRU problem). *Let $\mathcal{R}_q = \mathcal{R}^q_{D^+_N}$, χ_f be a distribution on $\mathcal{S}$, and χ_g be a distribution on $\mathcal{S}^-$.*

1. *The **decisional** structured-ANTRU or decisional s-ANTRU problem is to distinguish the s-$ANTRU_{\mathcal{R}_q,\chi_f,\chi_g}$ distribution from the uniform distribution on $\mathcal{S}^-$, given the same number of independent samples from both the distributions.*
2. *The **search** structured-ANTRU or search s-ANTRU problem is to find (f_1, f_2, g), given h from the s-ANTRU distribution.*

Assumption 1 (Decisional s-ANTRU assumption) *For any PPT adversary $\mathcal{A}$, the advantage to successfully solve the decisional s-ANTRU problem is negligible, i.e.,*

$$\mathsf{Adv}^{s-ANTRU}_{\mathcal{R}_q,\chi_f,\chi_g}(\mathcal{A}) = \big|\Pr[b = 1 \mid h \leftarrow s - ANTRU_{\mathcal{R}_q,\chi_f,\chi g} \;;\; b \leftarrow \mathcal{A}(h)] - \Pr[b = 1 \mid u \leftarrow_{\$} \mathcal{S}^- \;;\; b \leftarrow \mathcal{A}(u)]\big| \tag{15}$$

is negligible.

Definition 17 (s-RLWE distribution). *Let $\mathcal{R}_q = \mathcal{R}^q_{D^+_N}$, χ_r be distribution on $\mathcal{S}$, and χ_m is a distribution on $\mathcal{S}^-$. The structured-Ring RLWE or s-RLWE distribution is defined as*

$$s\text{-}RLWE_{\mathcal{R}_q,\chi_r,\chi_m} = \{(a, r_1 * a * r_2 + m \,(\mathrm{mod} q)) \mid r_1, r_2 \leftarrow \chi_r, m \leftarrow \chi_m, a \leftarrow_{\$} \mathcal{S}^-\} \tag{16}$$

Definition 18 (search s-RLWE problem). *Let $\mathcal{R}_q = \mathcal{R}^q_{D^+_N}$, χ_r be distribution on $\mathcal{S}$, and χ_m is a distribution on $\mathcal{S}^-$. The search structured-Ring LWE or search s-RLWE problem is to find (r_1, r_2, m), given $(a, r_1 * a * r_2 + m \,(\mathrm{mod}\, q)) \leftarrow s - RLWE_{\mathcal{R}_q,\chi_r,\chi_m}$.*

Assumption 2 (search s-RLWE assumption) *For any PPT adversary $\mathcal{A}$, the advantage to successfully solve the search s-RLWE problem is negligible, i.e.,*

$$\mathsf{Adv}^{s-RLWE}_{\mathcal{R}_q,\chi_r,\chi_m}(\mathcal{A}) = \Pr_{\substack{r_1,r_2\leftarrow\chi_r, m\leftarrow\chi_m \\ a\leftarrow_{\$}\mathcal{S}^-}}[(r_1, r_2, m) \leftarrow \mathcal{A}\,(a, r_1 * a * r_2 + m \,(\mathrm{mod} q))] \tag{17}$$

is negligible.

Theorem 1 (OW-CPA security). *Let $\mathcal{A}$ be a PPT adversary against the* OW-CPA *security of the scheme $\prod_{s-ANTRU.\mathsf{PKE}}$. Then, there exists an adversary $\mathcal{B}$ against the decisional s-ANTRU assumption and an adversary $\mathcal{C}$ against the search s-RLWE assumption, such that*

$$\mathsf{Adv}^{\mathsf{OW-CPA}}_{\prod_{s-ANTRU.\mathsf{PKE}}}(\mathcal{A}) \leq \mathsf{Adv}^{s-ANTRU}_{\mathcal{R}_q,\chi_f,\chi_g}(\mathcal{B}) + \mathsf{Adv}^{s-RLWE}_{\mathcal{R}_q,\chi_r,\chi_m}(\mathcal{C}) \tag{18}$$

where both $\mathcal{B}$ and $\mathcal{C}$ have roughly the same running time as $\mathcal{A}$.

Proof. Due to page limitations, the proof is provided in the supplementary material B. □

FO Transformation for IND-CCA KEM from OW-CPA PKE Applying standard transformations like the Fujisaki–Okamoto (FO) transformation [16,17], an OW-CPA secure PKE leads to an IND-CCA secure KEM. We instantiate our KEM following the FO variant described in Fig. 4. The resultant KEM is an instance of $\mathsf{KEM}_m^{\perp}$, discussed in [25], whose security is supported by a tight reduction in the ROM (random-oracle model) and a non-tight reduction in the QROM (quantum random oracle model) to the OW-CPA security of the underlying PKE, as stated in the following theorem adapted from [25,11].

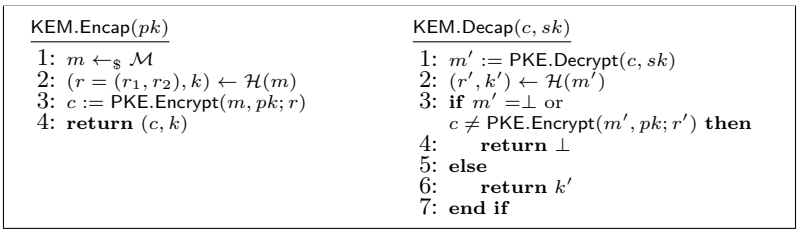

Fig. 4: General framework of Fujisaki-Okamoto transformation.

Theorem 2. *[25,11] Let* $\prod_{\mathsf{PKE}}$ *be a* δ*-correct and* γ *-spread public key encryption scheme. For any adversary* $\mathcal{A}$ *making at most* q_D *and* q_H *decapsulation and hash queries, respectively, against the* IND-CCA *security of the scheme* $\prod_{\mathsf{KEM}}$ *, there exists an adversary* $\mathcal{A}'$ *against the* OW-CPA *security of* $\prod_{\mathsf{PKE}}$ *such that*

- $\mathsf{Adv}^{\mathsf{IND-CCA}}_{\prod_{\mathsf{KEM}}}(\mathcal{A}) \leq 2q_H \cdot \mathsf{Adv}^{\mathsf{OW-CPA}}_{\prod_{\mathsf{PKE}}}(\mathcal{A}') + q_H 2^{-\gamma} + (2q_H + q_D)\cdot\delta$ *, with* $\mathsf{Time}(\mathcal{A}') \approx \mathsf{Time}(\mathcal{A})$ *, when the adversary has access to classical random oracles.*
- $\mathsf{Adv}^{\mathsf{IND-CCA}}_{\prod_{\mathsf{KEM}}}(\mathcal{A}) \leq 2q_T \sqrt{\mathsf{Adv}^{\mathsf{OW-CPA}}_{\prod_{\mathsf{PKE}}}(\mathcal{A}')} + 24q_T^2\sqrt{\delta} + 24\sqrt{q_T^3 q_D}\cdot 2^{-\gamma/4}$ *, with* $q_T = 2(q_D + q_H)$, $\mathsf{Time}(\mathcal{A}') \approx \mathsf{Time}(\mathcal{A}) + O(q_D \cdot q_H \mathsf{Time}(\mathsf{PKE.Encrypt}) + q_T^2)$ *, when* $\mathcal{A}$ *is a quantum adversary having access to quantum random oracles.*

Concrete Security Estimation Our analysis indicates that lattice attacks are not the most effective against the proposed scheme. We identify combinatorial search attacks as the primary threat and present **plausible** security estimation accordingly. While we believe our arguments are strong and resilient, in the unlikely event that an attacker reduces the problem to an NTRU-like equation, lattice attacks could become most effective. To address this possibility, we also consider **paranoid** security estimation, which assumes lattice attacks are the optimal attack strategy.

- **Plausible:** An attacker can opt for the following two approaches. ① Brute force search for $f_1, f_2 \in \mathcal{L}_f$, and check if $f_1 * h * f_2 \in \mathcal{L}_g$, which costs

$$|\mathcal{L}_f \times \mathcal{L}_f| \approx \binom{N+1}{\frac{d_f}{2}}^2 \binom{N+1-\frac{d_f}{2}}{\frac{d_f}{2}}^2.$$

Although a straightforward meet-in-the-middle attack does not seem to be applicable, we conservatively estimate the search space size to be $\sqrt{|\mathcal{L}_f \times \mathcal{L}_f|}$.

② Otherwise, solve a *Two-layer NTRU* problem. That is, first brute force search for $f_1 \in \mathcal{L}_f$ and since $f_1 * h = g * f_2^{-1} \pmod q$, which is again an NTRU-like problem over structured subsets of the twisted dihedral group. Therefore, the attacker needs to perform lattice reduction in a $4N$-dimensional lattice generated by the matrix

$$\begin{pmatrix} 1 & f_1 * h \\ 0 & q \end{pmatrix},$$

and for the correct guess recovers the elements $(g, f_2) \in \mathcal{L}_g \times \mathcal{L}_f$. The dimension of the lattice can be reduced to $2N$ due to the symmetry involved in the structure of the public key and the secrets. Therefore, this approach costs

$$|\mathcal{L}_f| \times \text{cost of lattice reduction in } 2N\text{-dimensional lattice.}$$

However, we very conservatively (favoring the attacker) assume that guessing f_1 correctly is a valid attack. Thus, reducing the cost of attack to just $|\mathcal{L}_f|$. It is important to emphasize that a meet-in-the-middle (MitM) attack is not applicable in this context, since $f_1 * h$ constitutes another NTRU instance rather than a short vector as in conventional MitM attacks. Nevertheless, for a conservative security estimation, we still consider the cost of the search attack to be $\sqrt{|\mathcal{L}_f|}$.

– **Paranoid:** In the worst case, an attacker would be able to successfully get an equation of the form $h' = f'^{-1} * g' \pmod q \in \mathcal{R}_q$ where $f' \approx f_1 * f_2$ and g' are short elements. Consequently, the secret (f', g') can be recovered by solving SVP in a $4N$-dimensional lattice generated by the matrix

$$\mathcal{B}_{h'}^{CS} = \begin{pmatrix} \mathbf{I}_{2N} & \mathbf{H}' \\ \mathbf{0}_{2N} & q\mathbf{I}_{2N} \end{pmatrix},$$

where $\mathbf{H}'$ is the matrix representation of h'. Moreover, for $h' \in \mathcal{S}^-$, the lattice dimension is reduced to $2N$. Taking into account both primal and hybrid lattice attacks, the final cost is determined by the minimum of these two methods.

Concrete security against lattice attacks is estimated based on the hardness of core-SVP [1] for BKZ-2.0 [8,2]. As mentioned earlier, the cost of BKZ with parameter β (known as blocksize) dominates the estimation. We utilized state-of-the-art estimators, specifically the Fatigue estimator [12] and the 2016-estimator [1], to estimate the required value of β. Consequently, the core-SVP cost of BKZ when combined with a sieving oracle is estimated to be $2^{0.292\beta+o(\beta)}$ (classical) [4] and $2^{0.265\beta+o(\beta)}$ (quantum) [35].

5 Parameter Selection

5.1 Analysis of Decryption Failure

The s-ANTRU encryption scheme requires the condition $4(pd_r^2 + d_f^2) < \frac{q}{2}$ for perfect correctness. However, by tolerating decryption failures within acceptable security thresholds, we can reduce the modulus q, thereby lowering bandwidth requirements. The following theorem gives the probability of decryption failure for s-ANTRU with the proof outlined in Appendix C.

Theorem 3. *For the s-ANTRU encryption scheme with the secrets $f_i \in \mathcal{L}_f, g \in \mathcal{L}_g$, randomnesses $r_i \in \mathcal{L}_r$, and the message $m \in \mathcal{L}_m$, the probability of decryption failure is estimated as*

$$P_{dec\_fail}^{N,q} = 4N \times \left(1 - \Phi\left(\frac{q}{2\tilde{\sigma}}\right)\right) < 2N \times \mathsf{erfc}\left(\frac{q}{2\sqrt{2}\tilde{\sigma}}\right), \tag{19}$$

where Φ is the cdf (cumulative distribution function) of the standard Normal distribution, erfc *is the complementary error function, and $\tilde{\sigma}^2 = \frac{4}{N}(p^2 d_r^2 d_g + d_f^2 d_m)$.*

5.2 Non Overstretched Parameters

Ducas and Woerdan [12], refining the work in [31], demonstrated that the BKZ algorithm performs better against NTRU lattices with a very large modulus q than previously estimated by the 2016-estimator [1]. This improvement is attributed to a phenomenon known as Dense Sublattice Discovery (DSD). Essentially, BKZ discovers a basis for the dense sublattice formed by the rotations of the short secret key prior to the actual Secret Key Recovery (SKR). This phenomenon is also observed in GR-NTRU lattices, which likewise contain a sublattice of half the dimension formed by the rotations of the secret key.

The term *Fatigue Point* is used for the modulus q value that separates the regular region and the overstretched region. Ducas and Woerdan established the following claim to provide an estimation of the Fatigue Point.

Theorem 4. *[12, Claim 3.5] The BKZ algorithm with blocksize $\beta = \mathcal{B}n$, applied to an NTRU instance with parameter $q = \Theta(n^{\mathcal{Q}})$ and secret $\|(f, g)\| = O(n^{\mathcal{P}})$, triggers the DSD event if*

$$\mathcal{B} = \frac{8\mathcal{P}}{\mathcal{Q}+1} + o(1). \tag{20}$$

We have replaced the notation $\mathcal{S}$ from the original statement in [12] with $\mathcal{P}$ because $\mathcal{S}$ is being used for a different purpose in this work. The relative blocksizes given by the 2016-estimate are $\frac{2\mathcal{Q}}{(\mathcal{Q}+1-\mathcal{P})^2}$ for $\mathcal{P} < 1$ and $\frac{2}{\mathcal{Q}+2-2\mathcal{P}}$ for $\mathcal{P} \geq 1$ [12]. Comparing the relative blocksize of the Fatigue estimation and the 2016-estimation yields the following conditions for the Fatigue Point:

$$q = n^{2.484+o(1)} \text{ for } \mathcal{P} < 1 \text{ and } q = n^{3.732+o(1)} \text{ for } \mathcal{P} \geq 1. \tag{21}$$

Empirical results in [12] for the ternary secret vectors or secrets sampled from a discrete Gaussian distribution with a mean of zero and a variance of $\frac{2}{3}$, in general, the norm of the secret is $O(\sqrt{N})$ (i.e., $\mathcal{P} = \frac{1}{2}$) provides a concrete bound $q \leq 0.004 \times n^{2.484}$ for parameter sets to be non overstretched. However, in our case, the norm of the secret is

$$\|(f', g')\| \approx \|(f_1 * f_2, g)\| = O(N), \text{ i.e., } \mathcal{P} = 1,$$

since the coefficients of $f_1 * f_2$ are normally distributed with mean 0 and variance $2d_f^2/N$, where $d_f = O(N)$. See Appendix C for more details on the distribution of coefficients. The experimental results indicate that our design parameters do not fall into the overstretched regime when $q \leq 0.00006 \times n^{3.732}$. It can be verified that the selected paranoid parameters in Table 4 are not overstretched.

5.3 Concrete Parameterization

Following the concrete security estimation, Tables 3 and 4 provide two parameter sets for ANTRU corresponding to the plausible and paranoid approaches, respectively, where N denotes half the order of the dihedral group, $d_f = d_g = d_r = d_m = d$, and $\mathcal{L} = \mathcal{T}(d, d) \cap \mathcal{S}$.

Table 3: Plausible parameter sets for ANTRU, targeting levels of security equivalent to AES 128, 192, and 256, respectively.

Security level	$(\boldsymbol{N}, \boldsymbol{d}, \boldsymbol{q})$	**Dec.** δ	$\sqrt{\lvert\mathcal{L}\rvert} = \sqrt{\binom{N+1}{d/2}\binom{N+1-d/2}{d/2}}$
128	(269, 84, 2^{13})	$< 2^{-132}$	2^{137}
192	(293, 120, 2^{14})	$< 2^{-196}$	2^{195}
256	(503, 132, 2^{14})	$< 2^{-256}$	2^{264}

Table 4: Paranoid parameter sets for ANTRU, targeting security levels comparable to those of NTRU-HPS in the third round of the NIST competition.

Core-SVP	$(\boldsymbol{N}, \boldsymbol{d}, \boldsymbol{q})$	**Classical**						**Quantum**					
		Primal		**Hybrid**				**Primal**		**Hybrid**			
		β	Cost	K	β	Reduction cost	Search cost	β	Cost	K	β	Reduction cost	Search cost
105	(379, 94, 2^{13})	386	113	207	357	104	**105**	386	**102**	202	382	101	102
144	(509, 146, 2^{14})	534	**156**	280	538	157	157	534	**142**	261	547	146	145
178	(653, 144, 2^{14})	680	199	375	618	179	**180**	680	180	348	626	**166**	166

Similarly, the parameters listed in Table 4 achieve negligible decryption failure rates of less than 2^{-128}, 2^{-192}, and 2^{-256}, in accordance with Theorem 3.

Computational cost The most computationally expensive operation in the scheme is multiplication within the dihedral group ring. Based on the discussion about multiplication between symmetric and antisymmetric elements in Section 3.4 and Appendix E, the cost of schoolbook multiplication for the constituent procedures in the reference implementation is presented below.

– KeyGen: The key generation procedure requires inverting two symmetric elements in $\mathcal{R}_q$ and $\mathcal{R}_p$, together with two multiplications between a symmetric

and an antisymmetric element. An inversion in $\mathcal{R}_q$ reduces to four multiplications and one inversion in $\mathbb{Z}_q C_N^+$ (see Algorithm 1). We implement constant-time inversion in $\mathbb{Z}_q C_N^+$ using the Bernstein-Yang (BY) algorithm [6]. For a prime power modulus $q = p^k$ (with $p = 2$ in our setting), the BY algorithm proceeds in two phases: it first computes the inverse of the input element in $\mathbb{Z}_p C_N^+$ at a linear cost in N, and then lifts this inverse to $\mathbb{Z}_q C_N^+$ by performing $2 \times \lceil \log_2 k \rceil$ polynomial multiplications. Consequently, the total cost of inverting a symmetric element in $\mathcal{R}_q$ is approximately $3(N+1)^2$, as $\log_2 k \leq 4$ for the parameters selected in Tables 3 and 4. Similarly, for $p = 3$, the cost of inversion in $\mathcal{R}_p$ is about $(N+1)^2$. Therefore, the overall computational cost of the KeyGen procedure can be approximated as $10N^2 + O(N)$.

- Encap: It involves two multiplications between a symmetric and antisymmetric element costing $2(N^2 - 1)$ integer multiplications.
- Decap: It involves six multiplications between a symmetric and antisymmetric element costing $6(N^2 - 1)$ integer multiplications.

A Note on Invertibility To maximize the likelihood that a randomly sampled $f \in \mathcal{L}_f$ is invertible in both $\mathcal{R}_p$ and $\mathcal{R}_q$, it suffices that its determinant $\det(f) \in \mathbb{Z}[x]/(x^N - 1)$ be invertible modulo p and modulo q with high probability. Since q is a power of 2, invertibility modulo q follows once $\det(f)$ is invertible modulo 2 (this lifts from $\mathbb{Z}_2$ to $\mathbb{Z}_{2^k}$ by standard Hensel-type arguments; see [45, Algorithm 3]). Empirically, the key space is very large: almost every random $f \in \mathcal{L}_f$ we sampled is invertible. A plausible explanation is that sampling $f \in \mathcal{T}(d_f + 1, d_f)$ ensures that

$$\det(f)(1) \neq 0 \pmod{2, 3, x^N - 1},$$

so $(x-1) \nmid \det(f)$ in $\mathbb{Z}_2[x]$ and $\mathbb{Z}_3[x]$. Hence $\gcd\big(\det(f), x^N - 1\big) = 1$ modulo 2 and 3 with high probability, which in turn implies that inverses modulo p and q exist with high probability. When N is prime we have

$$x^N - 1 = (x-1) * \Phi_N(x),$$

where $\Phi_N(x) = 1 + x + \cdots + x^{N-1}$ is the N-th cyclotomic polynomial. The probability of invertibility can be further improved by choosing a prime N for which 2 and 3 are primitive roots modulo N. In that case $\mathrm{ord}_N(2) = \mathrm{ord}_N(3) = N-1$, so $\Phi_N(x)$ is irreducible over both $\mathbb{Z}_2$ and $\mathbb{Z}_3$. Consequently, $\mathbb{Z}_2[x]/(\Phi_N(x))$ and $\mathbb{Z}_3[x]/(\Phi_N(x))$ are fields. Therefore, a random $f \in \mathcal{L}_f$ is invertible in $\mathcal{R}_p$ and $\mathcal{R}_q$ if and only if

$$\det(f) \neq 0 \pmod{2, 3, \Phi_N(x)}. \tag{22}$$

Equivalently, since $\deg\big(\det(f)\big) \leq N - 1 = \deg(\Phi_N(x))$ and Φ_N is irreducible over $\mathbb{Z}_2$ and $\mathbb{Z}_3$, condition (22) is the same as

$$\det(f) \neq \pm \Phi_N(x) \pmod{2, 3, x^N - 1}. \tag{23}$$

For a uniformly random $f \in \mathcal{L}_f$, the event that $\det(f)$ coincides with one of the nonzero scalar multiples of $\Phi_N(x)$ modulo 2 or 3 is negligible. Our experiments confirm that (23) holds with overwhelming probability. If desired, one can also enforce it deterministically by imposing mild constraints on the coefficients of f. Further details appear in Appendix H.

6 Future Scope

This work introduces a noncommutative analogue of NTRU that arguably mitigates the impact of lattice-based attacks against the construction and, hence, leads to more compact parameter sets. Our initial investigation invites potential extensions, which we briefly outline below, with a detailed discussion provided in Appendix I.

- **NTT for Faster Multiplications:** Further twisting the multiplication so that the resulting twisted dihedral group ring becomes isomorphic to

$$\mathbb{Z}_q^{\lambda} D_N \cong \frac{\mathbb{Z}_q[x,y]}{(x^N+1,\, y^2+1,\, xy - yx^{N-1})}$$

  enables faster multiplication in $\mathbb{Z}_q^{\lambda} D_N$, which is equivalent to four multiplications in the ring $\mathbb{Z}_q[x]/(x^N+1)$, using NTTs for suitable choices of N and q. However, this modification alters the matrix representation of the public key. Therefore, a more in-depth analysis is required to determine whether a homomorphism exists that could exploit this structure to achieve a dimension reduction. We leave this investigation for future work.
- **Digital Signatures and Challenges:** It is natural to explore the possibility of extending our approach beyond KEMs and constructing a digital signature scheme–for instance, a noncommutative analog of FALCON [14] that provides stronger resistance to lattice-based attacks. However, a major challenge in this direction lies in generating a trapdoor, which requires completing the basis over the noncommutative dihedral group ring. Specifically, given a short pair $(f, g) \in \mathbb{Z}D_N$, one must find another short pair $(F, G) \in \mathbb{Z}D_N$ such that $f * G - g * F = q$. To our knowledge, no method analogous to those available for cyclic group rings (as described in [14]) currently exists for the dihedral group ring. Consequently, substantial foundational work is required to realize this idea.
- **Smaller Modulus q:** The smaller the value of q, the lower the bandwidth required for establishing a shared key. Recent works, such as BAT [15] and DAWN [36], have achieved better conditions for decryption by either modifying the encryption process to eliminate the addition mask modulus p or by introducing small masking polynomials to the message and employing novel decoding methods for decryption. It would be beneficial to explore the possibility of integrating these ideas into our work to reduce communication costs. However, a straightforward adaptation appears to be nontrivial and would require significant analysis, so this will be addressed as future work.

References

1. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key {Exchange—A} new hope. In: 25th USENIX Security Symposium (USENIX Security 16). pp. 327–343 (2016), https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_alkim.pdf

2. Aono, Y., Wang, Y., Hayashi, T., Takagi, T.: Improved progressive BKZ algorithms and their precise cost estimation by sharp simulator. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 789–819. Springer (2016). https://doi.org/10.1007/978-3-662-49890-3_30
3. Bagheri, K., Sadeghi, M.R., Panario, D.: A Non-commutative Cryptosystem Based on Quaternion Algebras. Designs, Codes and Cryptography **86**, 2345–2377 (10 2018). https://doi.org/10.1007/s10623-017-0451-4
4. Becker, A., Ducas, L., Gama, N., Laarhoven, T.: New directions in nearest neighbor searching with applications to lattice sieving. In: Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms. pp. 10–24. SIAM (2016). https://doi.org/10.1137/1.9781611974331.ch2
5. Bernstein, D.J., Chuengsatiansup, C., Lange, T., van Vredendaal, C.: NTRU Prime: Reducing Attack Surface at Low Cost. In: Selected Areas in Cryptography – SAC 2017. pp. 235–260. Springer International Publishing (2018). https://doi.org/10.1007/978-3-319-72565-9_12
6. Bernstein, D.J., Yang, B.Y.: Fast constant-time gcd computation and modular inversion. IACR Transactions on Cryptographic Hardware and Embedded Systems **2019**(3), 340–398 (May 2019). https://doi.org/10.13154/tches.v2019.i3.340-398
7. Chen, C., Danba, O., Hoffstein, J., H¨ulsing, A., Rijneveld, J., Schanck, J.M., Saito, T., Schwade, P.S., Whyte, W.W., Xagawa, K.X., Yamakawa, T., Zhang, Z.: PQC round-3 candidate: NTRU. technical report. Tech. rep., NTRU Cryptosystems Technical Report No.11, Version 2, March 2001. Report (2019), https://ntru.org/f/ntru-20190330.pdf
8. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 1–20. Springer (2011). https://doi.org/10.1007/978-3-642-25385-0_1
9. Coppersmith, D.: Attacking non-commutative NTRU. Tech. rep., Technical report, IBM research report, April 1997. Report (2006), https://dominoweb.draco.res.ibm.com/d102d0885e971b558525659300727a26.html
10. Coppersmith, D., Shamir, A.: Lattice Attacks on NTRU. In: Advances in Cryptology — EUROCRYPT '97. pp. 52–61. Springer Berlin Heidelberg, Berlin, Heidelberg (1997). https://doi.org/10.1007/3-540-69053-0_5
11. Don, J., Fehr, S., Majenz, C., Schaffner, C.: Online-Extractability in the Quantum Random-Oracle Model. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022. pp. 677–706. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-07082-2_24
12. Ducas, L., van Woerden, W.: NTRU Fatigue: How Stretched is Overstretched? In: Advances in Cryptology – ASIACRYPT 2021. pp. 3–32. Springer International Publishing (2021). https://doi.org/10.1007/978-3-030-92068-5_1
13. Esser, A., Girme, R., Mukherjee, A., Sarkar, S.: Memory-efficient attacks on small lwe keys. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023. pp. 72–105. Springer Nature Singapore, Singapore (2023). https://doi.org/10.1007/978-981-99-8730-6_3
14. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhan, Z.: FALCON: Fast-Fourier lattice-based compact signatures over NTRU. Tech. rep., National Institute of Standards and Technology (NIST) (2018), https://www.di.ens.fr/~prest/Publications/falcon.pdf

15. 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**(2), 240–265 (Feb 2022). https://doi.org/10.46586/tches.v2022.i2.240-265, https://tches.iacr.org/index.php/TCHES/article/view/9487
16. Fujisaki, E., Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes. In: Wiener, M. (ed.) Advances in Cryptology — CRYPTO' 99. pp. 537–554. Springer Berlin Heidelberg, Berlin, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_34
17. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology **26**, 80–101 (2013), https://doi.org/10.1007/s00145-011-9114-1
18. Gärtner, J.: NTWE: A Natural Combination of NTRU and LWE. In: Johansson, T., Smith-Tone, D. (eds.) Post-Quantum Cryptography (PQCrypto 2023). pp. 321–353. Springer Nature Switzerland, Cham (2023), https://doi.org/10.1007/978-3-031-40003-2_12
19. Genise, N., Gentry, C., Halevi, S., Li, B., Micciancio, D.: Homomorphic encryption for finite automata. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology – ASIACRYPT 2019. pp. 473–502. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-34621-8_17
20. Gentry, C.: Key recovery and message attacks on NTRU-composite. In: Pfitzmann, B. (ed.) Advances in Cryptology — EUROCRYPT 2001. pp. 182–194. Springer Berlin Heidelberg, Berlin, Heidelberg (2001), https://doi.org/10.1007/3-540-44987-6_12
21. Hoffstein, J., Howgrave-Graham, N., Pipher, J., Silverman, J.H., Whyte, W.: NTRUSign: Digital Signatures Using the NTRU Lattice. In: Joye, M. (ed.) Topics in Cryptology — CT-RSA 2003. pp. 122–140. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
22. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: International algorithmic number theory symposium. pp. 267–288. Springer, Berlin, Heidelberg (1998). https://doi.org/10.1007/BFb0054868
23. Hoffstein, J., Silverman, J.H.: A non-commutative version of the NTRU public key cryptosystem. unpublished paper, February (1997)
24. Hoffstein, J., Silverman, J.H., Whyte, W.: Meet-in-the-middle attack on an NTRU private key. Tech. rep., Technical report, NTRU Cryptosystems, July 2006. Report (2006), https://ntru.org/f/tr/tr004v2.pdf
25. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A Modular Analysis of the Fujisaki-Okamoto Transformation. In: Kalai, Y., Reyzin, L. (eds.) Theory of Cryptography. pp. 341–371. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-70500-2_12
26. Howgrave-Graham, N.: A Hybrid Lattice-Reduction and Meet-in-the-Middle Attack Against NTRU. In: Advances in Cryptology - CRYPTO 2007. pp. 150–169. Springer Berlin Heidelberg (2007). https://doi.org/10.1007/978-3-540-74143-5_9
27. Hülsing, A., Rijneveld, J., Schanck, J.M., Schwabe, P.: NTRU-HRSS-KEM: Algorithm specifications and supporting documentation. NIST submission **30** (2017), https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions
28. Hurley, T.: Group rings and rings of matrices. International Journal of Pure and Applied Mathematics **31**, 319–335 (01 2006), https://www.researchgate.net/publication/228928727_Group_rings_and_rings_of_matrices

29. Kannan, R.: Improved algorithms for integer programming and related lattice problems. In: Proceedings of the fifteenth annual ACM symposium on Theory of computing. pp. 193–206 (1983), https://doi.org/10.1145/800061.808749
30. 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, https://sites.google.com/view/ntruplus
31. Kirchner, P., Fouque, P.A.: Revisiting Lattice Attacks on Overstretched NTRU Parameters. In: Advances in Cryptology – EUROCRYPT 2017. pp. 3–26. Springer International Publishing (2017). https://doi.org/10.1007/978-3-319-56620-7_1
32. Kumar, V., Raya, A., Gangopadhyay, A.K., Gangopadhyay, S.: Dimension Reduction Attack on Noncommutative Group Ring NTRU Over the Dihedral Group. In: 2024 1st International Conference On Cryptography And Information Security (VCRIS). pp. 1–6 (2024). https://doi.org/10.1109/VCRIS63677.2024.10813443
33. Kumar, V., Raya, A., Gangopadhyay, A.K., Gangopadhyay, S., Hussain, M.T.: An Efficient Noncommutative NTRU from Semidirect Product. In: Mukhopadhyay, S., Stănică, P. (eds.) Progress in Cryptology – INDOCRYPT 2024. pp. 3–27. Springer Nature Switzerland, Cham (2025). https://doi.org/10.1007/978-3-031-80308-6_1
34. Kumar, V., Raya, A., Gangopadhyay, S., Gangopadhyay, A.K.: Cryptanalysis of Group Ring NTRU: The Case of the Dihedral Group. Security and Privacy **8**, 1–15 (2025), https://doi.org/10.1002/spy2.70020
35. Laarhoven, T.: Search problems in cryptography: from fingerprinting to lattice sieving. Phd thesis, Eindhoven University of Technology (2015), https://research.tue.nl/en/publications/search-problems-in-cryptography-from-fingerprinting-to-lattice-si
36. Liu, Y., Zhang, Y., Lu, X., Cheng, Y., Yin, Y.: DAWN: Smaller and faster NTRU encryption via double encoding. Cryptology ePrint Archive, Paper 2025/1520 (2025), https://eprint.iacr.org/2025/1520
37. 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, https://tches.iacr.org/index.php/TCHES/article/view/8293
38. Malekian, E., Zakerolhosseini, A., Mashatan, A.: QTRU : a lattice attack resistant version of NTRU PKCS based on quaternion algebra. IACR Cryptology ePrint Archive **386** (2009), https://eprint.iacr.org/2009/386
39. May, A.: How to meet ternary lwe keys. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology – CRYPTO 2021. pp. 701–731. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-84245-1_24
40. Raya, A., Kumar, V., Gangopadhyay, A.K., Gangopadhyay, S.: Giant Does NOT Mean Strong: Cryptanalysis of BQTRU. In: Niederhagen, R., Saarinen, M.J.O. (eds.) Post-Quantum Cryptography. pp. 312–348. Springer Nature Switzerland, Cham (2025). https://doi.org/10.1007/978-3-031-86599-2_11
41. Raya, A., Kumar, V., Gangopadhyay, S.: DiTRU: A Resurrection of NTRU over Dihedral Group. In: Vaudenay, S., Petit, C. (eds.) Progress in Cryptology - AFRICACRYPT 2024. pp. 349–375. Springer Nature Switzerland, Cham (2024), 10.1007/978-3-031-64381-1_16
42. Raya, A., Kumar, V., Gangopadhyay, S., Gangopadhyay, A.K.: Efficient Key Encapsulation Mechanisms from Noncommutative NTRU. Computer Networks

272, 111704 (2025). https://doi.org/https://doi.org/10.1016/j.comnet.2025.111704
43. Schnorr, C.P.: A hierarchy of polynomial time lattice basis reduction algorithms. Theoretical computer science **53**(2-3), 201–224 (1987). https://doi.org/10.1016/0304-3975(87)90064-8
44. Schwabe, P., Avanzi, R., Bos, J.B., Ducas, L.D., Kiltz, E., crede Lepoint, T., Lyubashevsky, V., M. Schanck, J., Seiler, G., Stehle, D.: Crystals-Kyber. Tech. rep., National Institute of Standards and Technology (NIST) (2017), https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions
45. Silverman, J.H.: Almost inverses and fast NTRU key creation. NTRU Cryptosystems Technical Report #14 (1999), https://ntru.org/f/tr/tr014v1.pdf
46. Truman, K.R.: Analysis and Extension of Non-Commutative NTRU. PhD dissertation, University of Maryland (2007), https://drum.lib.umd.edu/handle/1903/7344
47. Yasuda, T., Dahan, X., Sakurai, K.: Characterizing NTRU-variants using group ring and evaluating their lattice security. IACR Cryptol. ePrint Arch. p. 1170 (2015), http://eprint.iacr.org/2015/1170

A Terminologies and Security Notions

A.1 Public Key Encryption

A **P**ublic **K**ey **E**ncryption scheme or a PKE with the message space $\mathcal{M}$, ciphertext space $\mathcal{C}$, and the key space $\mathcal{K}$ is a tuple of three PPT (probabilistic polynomial time) algorithms $(\mathsf{PKE.KeyGen}, \mathsf{PKE.Encrypt}, \mathsf{PKE.Decrypt})$ where

- $\mathsf{PKE.KeyGen}(1^\lambda)$ is the key generation algorithm that outputs a secret and public key pair (sk, pk) according to the input parameter λ that corresponds to the security level, denoted as $(sk, pk) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$.
- $\mathsf{PKE.Encrypt}(m, pk; r)$ outputs the ciphertext corresponding to the input message $m \in \mathcal{M}$, the public key pk, and a possible random string r, denoted as $c = \mathsf{PKE.Encrypt}(m, pk; r)$.
- $\mathsf{PKE.Decrypt}(c, sk)$ is the decryption algorithm that outputs a message $m' \in \mathcal{M}$ or a symbol $\perp \notin \mathcal{M}$ (when $c \notin \mathcal{C}$) on input a ciphertext $c \in \mathcal{C}$ and the secret key sk, denoted as $m' = \mathsf{PKE.Decrypt}(c, sk)$.

Correctness: For $0 \leq \delta < 1$, a PKE is δ-correct if for every key pair (sk, pk),

$$\Pr[\mathsf{PKE.Decrypt}(\mathsf{PKE.Encrypt}(m, pk), sk)] \neq m \leq \delta,$$

in the worst-case, where the probability is taken over all the randomness used in PKE.KeyGen and PKE.Encrypt. For $\delta = 0$, we say that the PKE is perfectly correct. We say that a PKE is γ-spread if

$$\min_{\substack{m \in \mathcal{M} \\ (sk, pk)}} \left(-\log \max_{c \in \mathcal{C}} \Pr[c = \mathsf{Encrypt}(m, pk, r)] \right) \geq \gamma.$$

Definition 19 (OW/IND-CPA PKE). *A* PKE *scheme* $\prod_{\mathsf{PKE}}$ *is* OW/IND-CPA *secure if for any PPT adversary* $\mathcal{A}$*, the advantages* $\mathsf{Adv}^{\mathsf{OW-CPA}}_{\prod_{\mathsf{PKE}}}(\mathcal{A}) := \Pr[m' = m]$ *and* $\mathsf{Adv}^{\mathsf{IND-CPA}}_{\prod_{\mathsf{PKE}}}(\mathcal{A}) := |\Pr[b' = b] - 1/2|$ *defined in the security games in Figure 5 are negligible.*

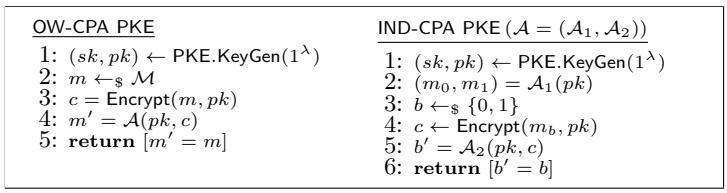

Fig. 5: OW/IND-CPA security game for the scheme $\prod_{\mathsf{PKE}}$.

A.2 Key Encapsulation Mechanism

A **K**ey **E**ncapsulation **M**echanism or a KEM is a tuple consisting of three PPT algorithms (KEM.KeyGen, KEM.Encap, KEM.Decap) that facilitates the sharing of a secret key between two parties.

- $\mathsf{KEM.KeyGen}(1^\lambda)$ is the key generation algorithm that outputs a secret and public key pair (sk, pk) according to the input parameter λ that corresponds to the security level, denoted as $(sk, pk) \leftarrow \mathsf{KEM.KeyGen}(1^\lambda)$.
- $\mathsf{KEM.Encap}(pk)$ is the encapsulation algorithm[4] that outputs an encapsulated ciphertext $c \in \mathcal{C}$ and the associated session key $k \in \mathcal{K}$ on input a public key pk, denoted as $(c, k) = \mathsf{KEM.Encap}(pk)$.
- $\mathsf{KEM.Decap}(c, sk)$ is the decapsulation algorithm that outputs the session k on input an encapsulated ciphertext c and a secret key sk, denoted as $k = \mathsf{KEM.Decap}(c, sk)$.

Correctness: For $0 \leq \delta < 1$, a KEM is δ-correct if for every key pair (sk, pk),

$$\Pr[\mathsf{KEM.Decap}(c, sk) \neq k \mid \mathsf{KEM.Encap}(pk) = (c, k)] \leq \delta,$$

where the probability is taken over all the randomness used in KEM.KeyGen and KEM.Encap. For $\delta = 0$, we say that the KEM is perfectly correct.

Definition 20 (IND-CCA KEM). *A* KEM *scheme* $\prod_{\mathsf{KEM}}$ *is called* IND-CCA *secure if the advantage* $\mathsf{Adv}^{\mathsf{IND-CCA}}_{\prod_{\mathsf{KEM}}}(\mathcal{A}) := \left|\Pr[b_i = b] - \frac{1}{2}\right|$ *of a PPT adversary* $\mathcal{A}$ *in the security game in Figure 6 is negligible.*

[4] KEM.Encap is a randomized algorithm that, everytime produces a fresh encapsulated key–ciphertext pair determined by its internally sampled randomness.

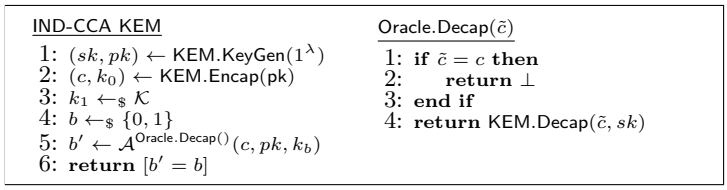

Fig. 6: IND-CCA security game for the scheme $\prod_{\mathsf{KEM}}$.

B Proof of Theorem 1

We consider the following sequence of games G_i to prove the OW-CPA of our scheme $\prod_{s-ANTRU.\mathsf{PKE}}$. Let $\mathsf{Adv}_{G_i}^{\mathsf{OW-CPA}}$ denotes the probability of winning the game G_i.

Game G_0: This is the genuine OW-CPA game as shown in Figure 5. Then, by definition $\mathsf{Adv}_{G_0}^{\mathsf{OW-CPA}} = \mathsf{Adv}^{\mathsf{OW-CPA}}_{\prod_{s-ANTRU.\mathsf{PKE}}}(\mathcal{A})$.

Game G_1: This game is identical to game G_0 except that the public key h is replace by an element a sampled from $U(\mathcal{S}^-)$. Using the decisional s-ANTRU assumption, we have

$$\left|\mathsf{Adv}_{G_0}^{\mathsf{OW-CPA}} - \mathsf{Adv}_{G_1}^{\mathsf{OW-CPA}}\right| \leq \mathsf{Adv}^{s-ANTRU}_{\mathcal{R}_q,\chi_f,\chi_g}(\mathcal{B})$$

Game G_2: This game is identical to game G_1 except that random element $a \in \mathcal{S}^-$ in game G_1 is replaced by $a/p \in \mathcal{S}^-$, where p is coprime to q. Since a is sampled uniformly at random from the set $\mathcal{S}^-$ and $gcd(p, q) = 1$, therefore, the element $p \in \mathbb{Z}_q$ is invertible, hence multiplication by $p^{-1} \pmod q$ defines a bijection on $\mathcal{S}^-$. In fact, it is a bijection on $\mathcal{R}_q$. Therefore, the distribution of $a' = p^{-1}a \pmod q$ is also uniform over $\mathcal{S}^-$. Equivalently, if a is uniform in $\mathcal{S}^-$, then so is a/p. As a result

$$\mathsf{Adv}_{G_1}^{\mathsf{OW-CPA}} = \mathsf{Adv}_{G_2}^{\mathsf{OW-CPA}}$$

Moreover, under the search s-RLWE assumption, we have

$$\mathsf{Adv}_{G_2}^{\mathsf{OW-CPA}} = \mathsf{Adv}^{s-RLWE}_{\mathcal{R}_q,\chi_r,\chi_m}(\mathcal{C}).$$

Consequently,

$$\mathsf{Adv}^{\mathsf{OW-CPA}}_{\prod_{s-ANTRU.\mathsf{PKE}}}(\mathcal{A}) \leq \mathsf{Adv}^{s-ANTRU}_{\mathcal{R}_q,\chi_f,\chi_g}(\mathcal{B}) + \mathsf{Adv}^{s-RLWE}_{\mathcal{R}_q,\chi_r,\chi_m}(\mathcal{C}).$$

C Probability of Decryption Failure (Proof of Theorem 3)

The ciphertext equation is given by

$$c = p(r_1 * h * r_2) + m \pmod q. \tag{24}$$

For decryption, the receiver performs

$$f_1 * c * f_2 = p(r_1 * g * r_2) + f_1 * m * f_2 \pmod q. \tag{25}$$

Therefore, the correct decryption requires

$$\|p(r_1 * g * r_2) + f_1 * m * f_2\|_\infty < q/2.$$

We have $r_1, r_2 \in \mathcal{L}_r$, $g \in \mathcal{L}_g$, $f_1, f_2 \in \mathcal{T}(d_f, d_f) \cap \mathcal{S}$ (instead of $\mathcal{L}_f$ for simplicity), and $m \in \mathcal{L}_m$. Let us assume that $d_f \leq d_r \leq d_g \leq d_m$. Then,

$$\|p(r_1 * g * r_2) + f_1 * m * f_2\|_\infty \leq 4(pd_r^2 + d_f^2).$$

Consequently, perfect correctness can be achieved if

$$4(pd_r^2 + d_f^2) < q/2.$$

Therefore, even for very sparse f_i, r_i, allowing no decryption failure results in a significantly larger value for the modulus q, which consequently increases the sizes of the public key and ciphertext. However, by permitting negligible decryption failures, in line with NIST guidelines, it is possible to choose smaller values for q.

We analyze the distribution of coefficients of $f_1 * m * f_2$, and similar results will hold for $r_1 * g * r_2$. Let U_k denotes the kth coefficient of $u = f_1 * m$, then U_k is the sum of $2N$ terms of the form

$$U_k = \sum_{i,j} (f_1)_i (m)_j \text{ (summing over } 2N \text{ pairs of } i, j \text{ corresponding to } k).$$

For every pair (i, j),

$$E[(f_1)_i (m)_j] = 0 \text{ and } Var[(f_1)_i (m)_j] = \frac{d_f d_m}{N^2}.$$

Hence,

$$E[U_k] = 0 \text{ and } Var[U_k] = \frac{2d_f d_m}{N}.$$

By the central limit theorem, for large N,

$$U_k \sim N(0, \sigma_1^2) \text{ where } \sigma_1^2 = \frac{2d_f d_m}{N}.$$

Let $v = u * f_2$, and the random variable V_k denotes the kth coefficient of v. Then, again V_k is the sum of $2N$ terms of the form

$$V_k = \sum_{i,j} u_i (f_2)_j \text{ (summing over } 2N \text{ pairs of } i, j \text{ corresponding to } k).$$

Since f_2 takes exactly $2d_f$ nonzero values equal to ± 1, therefore, V_k is the sum of $2d_f$ random variables following $N(0, \sigma_1^2)$. As a result,

$$V_k \sim N(0, 2d_f \sigma_1^2). \tag{26}$$

Similarly, let the random variable W_k denotes the kth coefficient of $r_1 * g * r_2$ then

$$W_k \sim N(0, 2d_r \sigma_2^2) \tag{27}$$

where $\sigma_2^2 = 2d_r d_g/N$. For ease of calculations, let us assume that g and m follow same distribution, i.e., $d_g = d_m$, and r_i and f_i follow same distribution, i.e., $d_r = d_f$. This gives $\sigma_1^2 = \sigma_2^2 = \sigma^2$ (let), and $W_k \sim V_k$. Now, let us define a random variable Y_k as

$$Y_k = pW_k + V_k,$$

i.e., Y_k denotes the kth coefficient of $p(r_1 * g * r_2) + f_1 * m * f_2$. We have

$$Y_k \sim N(0, 2d_f(p^2+1)\sigma^2), \tag{28}$$

with

$$E[Y_k] = 0 \text{ and } Var[Y_k] = 2d_f\sigma^2(p^2+1) \ (= \tilde{\sigma}^2, \text{ let}). \tag{29}$$

For $q > 0$, consider

$$\begin{aligned} \Pr[|Y_k| > q/2] &= 2(1 - \Pr[Y_k \leq q/2]) \\ &= 2\left(1 - \Phi_{\tilde{\sigma}}\left(\frac{q}{2}\right)\right) \\ &= 2\left(1 - \Phi\left(\frac{q}{2\tilde{\sigma}}\right)\right) \end{aligned} \tag{30}$$

where $\Phi_{\tilde{\sigma}}(z)$ is the cdf of $N(0, \tilde{\sigma}^2)$, and $\Phi_{\tilde{\sigma}}(z) = \Phi(z/\tilde{\sigma})$, where Φ is the cdf of standard normal distribution $N(0,1)$.

For a given $q > 0$, to compute the probability of decryption failure, we need to compute the probability of the event when any coefficient of $p(r_1 * g * r_2) + f_1 * m * f_2$ exceeds $q/2$, which is equal to

$$\begin{aligned} P_{dec\_fail}^{N,q} &= 2N * \Pr[|Y_k| > q/2] \\ &= 4N * \left(1 - \Phi\left(\frac{q}{2\tilde{\sigma}}\right)\right) \qquad (31) \\ &< 2N * \mathsf{erfc}\left(\frac{q}{2\sqrt{2}\tilde{\sigma}}\right), \qquad (32) \end{aligned}$$

where erfc is the complementary error function.

D Hoffstein and Silverman Construction

This section describes the first construction of noncommutative NTRU by Silverman and Hoffstein [23,46], intended to avoid lattice attacks.

Setup For a dihedral group of order $2N$ defined as $D_N = \langle x, y \mid x^N = y^2 = 1, xy = yx^{N-1} \rangle$, let $\mathcal{R}$ indicate the group ring of the dihedral group and the integer ring

$$\mathcal{R} = \mathbb{Z}D_N \approx \frac{\mathbb{Z}[x,y]}{(x^N - 1, y^2 - 1, xy - yx^{N-1})}. \tag{33}$$

Furthermore, the cryptosystem needs a commutative subring $\mathcal{S}$,

$$\mathcal{S} = \{f \in \mathcal{R} \mid fy = yf\}. \tag{34}$$

The subring contains all the elements in $\mathcal{R}$ that commutes with y, i.e., $\overline{f(x)} = f(x)$ for all $f(x)$ in $\mathcal{S}$. Furthermore, the cryptosystem involves the following moduli: $p, q, r, s, t \in \mathbb{N}$, where p, r, and t are considered small, while q and s are large. We assume that p and q are prime and are coprime with $2N$. The cryptosystem uses the following rings and subrings:

$$\begin{aligned} &\mathcal{R}_q = \mathbb{Z}_q D_N, && \mathcal{R}_s = \mathbb{Z}_s D_N, && \mathcal{R}_r = \mathbb{Z}_r D_N, \\ &\mathcal{S}_q = \{f \in \mathcal{R}_q \mid fy = yf\}, && \mathcal{S}_s = \{f \in \mathcal{R}_s \mid fy = yf\}, && \mathcal{S}_r = \{f \in \mathcal{R}_r \mid fy = yf\}. \end{aligned}$$

Further, let us consider the following sample spaces:

$$\mathcal{L}_f = \mathcal{S}_q,\ \mathcal{L}_m = \mathcal{R}_q,\ \mathcal{L}_\phi = \mathcal{S}_r,\ \mathcal{L}_\psi = \mathcal{S}_s, \text{ and } \mathcal{L}_\omega = \mathcal{R}_t.$$

$\underline{\mathsf{KeyGen}(1^\lambda)}$
1: initialize $\mathsf{Sampler}$ with $\mathcal{L}_f$ and seed_f
2: do $f \leftarrow \mathsf{Sampler}$ until f is invertible in $\mathcal{R}_q$
3: initialize $\mathsf{sampler}$ with $\mathcal{L}_\omega$ and seed_ω
4: do $\omega \leftarrow \mathsf{Sampler}$
5: $F = f^{-1} \pmod q$
6: $h = pF * \omega * f \pmod q$
7: $pk = h, sk = (f, F, \omega)$
8: **return** (sk, pk)

$\underline{\mathsf{Encrypt}(m, h, \mathsf{seed}_{\phi,\phi',\psi})}$
1: initialize $\mathsf{Sampler}$ with $\mathcal{L}_\phi$ and seed_ϕ
2: $\phi \leftarrow \mathsf{Sampler}$
3: initialize $\mathsf{Sampler}$ with $\mathcal{L}_\phi$ and $\mathsf{seed}_{\phi'}$
4: $\phi' \leftarrow \mathsf{Sampler}$
5: initialize $\mathsf{Sampler}$ with $\mathcal{L}_\psi$ and seed_ψ
6: $\psi \leftarrow \mathsf{Sampler}$
7: $\Psi = \psi \pmod p$
8: $e = \phi * h * \phi' + \psi \pmod q$
9: $E = \Psi * h + m \pmod q$
10: **return** (e, E)

$\underline{\mathsf{Decrypt}((e, E), (f, F, \omega))}$
1: $a = f * e * F = p\phi * \omega * \phi' + \psi \pmod q$
2: $a' = \mathsf{Centerlift}(a, q)$
3: $\Psi = a' \pmod p$
4: $m' = E - \Psi * h \pmod q$
5: **return** m'

Fig. 7: A sketched description of the noncommutative NTRU scheme by Hoffstein and Silverman.

Coppersmith's Attack An initial analysis of the security of the Hoffstein-Silverman design argued that it resists lattice-based attacks based on the following points:

(i) The public key $h = pF * \omega * f \pmod q$ is no longer a product of polynomials with small norms. As a result, attacking the public key cannot be directly reduced to solving the Shortest Vector Problem (SVP) in certain lattices.
(ii) In the first component of the ciphertext pair, $e = \phi * h * \phi' + \psi \pmod q$, the noise term ψ does not have small coefficients. Furthermore, due to the noncommutative nature of the multiplication, the unknowns ϕ and ϕ' cannot be combined into a single side of the expression. Consequently, formulating an attack as an instance of the Closest Vector Problem (CVP) is not feasible against e.
(iii) In the second component of the ciphertext pair, $E = \Psi * h + m \pmod q$, the message m does not consist of small coefficients. Therefore, lattice-based attacks via CVP are similarly ineffective against E.

However, Coppersmith [9] introduced an attack against this design and showed that a third party that does not know the private key can decrypt the message. The attack depends on the following observations:

(i) **Retrieving ω' with Small Coefficients:** Since the attacker knows h, they can construct the element $h + yhy \in \mathcal{S}$[5]

$$\begin{aligned} h + yhy &= pF(\omega + y\omega y) * f \pmod q \\ &= pF * f * (\omega + y\omega y) \pmod q \\ &= p(\omega + y\omega y) \pmod q \end{aligned} \tag{35}$$

The second line of Equation (35) follows from the fact that $\omega + y\omega y \in \mathcal{S}$, and therefore commutes with both f and F. Moreover, since $\omega \in \mathcal{L}_\omega = \mathcal{R}_t$ for small t, the coefficients of $\omega + y\omega y \pmod q$ remain small and do not wrap around modulo q. Hence, the attacker can construct $\omega' \in \mathcal{S}_w$ with small coefficients satisfying that

$$h + yhy = p(\omega' + y\omega' y) \pmod q. \tag{36}$$

Building such ω' is possible since:

$$\begin{aligned} \omega + y\omega y &= (\omega_0 + y\omega_1) + y(\omega_0 + y\omega_1)y \\ &= (\omega_0 + \overline{\omega_0}) + y(\omega_1 + \overline{\omega_1}) \\ &= \left(2w_{00} + \sum_{i=1}^{N-1} (\omega_{0i} + \omega_{0N-i})x^i\right) + y\left(2\omega_{10} + \sum_{i=1}^{N-1} (\omega_{1i} + \omega_{1N-i})x^i\right). \end{aligned} \tag{37}$$

The coefficients of ω_{0i}, ω_{1i} lie in $\mathbb{Z}_t$, and therefore the coefficients of $h' = h'_0 + yh'_1 = h + yhy \pmod q$ belong to the set $\{-p(t-1), -p(t-2), \ldots, p(t-2), p(t-1)\}$.
For each even coefficient of h'_0 written as $2kp$, one can assign $\omega'_{0i} = \omega'_{0N-i} = k$. For each odd coefficient written as $(2k+1)p$, one can assign $\omega'_{0i} = k+1$ and $\omega'_{0N-i} = k$. The same procedure is applied to the coefficients of ω'_1 using the corresponding coefficients from h'_1. Hence, we can construct ω' with small coefficients without knowledge of the secret key.

(ii) **Defining a Special-Purpose Map θ:** The legitimate decryption process relies on applying a specific map that enables the recovery of Ψ from the first ciphertext component e, given knowledge of both f and F (the secret key).
In contrast, the attack strategy introduces a map θ designed to extract Ψ from the ciphertext component e *without* requiring knowledge of the secret components (f, F).
Before defining θ, we introduce the subset $\mathcal{S}^- \subset \mathcal{R}$ as:

$$\mathcal{S}^- = \{f \in \mathcal{R} \mid fy = -yf\}.$$

[5] **Caution:** For any element $h \in \mathbb{Z}D_N$, $h + yhy \in \mathcal{S}, h - yhy \in \mathcal{S}^-$. However, for an element in $h \in \mathbb{Z}D_N^+$ (twisted), $h + yhy \in \mathcal{S}^-, h - yhy \in \mathcal{S}$.

The map θ is a linear transformation:

$$\theta : \mathcal{R}_q \mapsto \mathcal{R}_q,$$

with the following properties:

- acts as the identity on $\mathcal{S}$.
- maps any $f_1 \in \mathcal{S}^-$ to some $f_2 \in \mathcal{S}^-$.
- is left- and right-linear with respect to $\mathcal{S}$.
- satisfies $\theta(h) = p\omega'$, where $\omega' \pmod q$ has small coefficients (as discussed earlier).

To construct such a map θ, we first express h and $p\omega'$ as:

$$h = h_0 + yh_1 = (h_{0s} + h_{0s^-}) + y\,(h_{1s} + h_{1s^-}) = \underbrace{h_{0s} + yh_{1s}}_{h_s \in \mathcal{S}} + \underbrace{h_{0s^-} + yh_{1s^-}}_{h_{s^-} \in \mathcal{S}^-},$$

$$p\omega' = p\omega'_0 + yp\omega'_1 = \left(\omega'_{0s} + \omega'_{0s^-}\right) + y\left(\omega'_{1s} + \omega'_{1s^-}\right) = \underbrace{\omega'_{0s} + y\omega'_{1s}}_{w'_s \in \mathcal{S}} + \underbrace{\omega'_{0s^-} + y\omega'_{1s^-}}_{w'_{s^-} \in \mathcal{S}^-}.$$

Any element $f \in \mathcal{S}$ can be written as:

$$f = a_0 + yb_0 + \sum_{i=1}^{(N-1)/2} \left(c_i(x^i + x^{N-i}) + yd_i(x^i + x^{N-i})\right),$$

and any $g \in \mathcal{S}^-$ can be written as:

$$g = \sum_{i=1}^{(N-1)/2} \left(c'_i(x^i - x^{N-i}) + yd'_i(x^i - x^{N-i})\right).$$

We then set $\omega'_s = h_s$ and $\omega'_{s^-} = \theta(h_{s^-})$. Coppersmith showed that in the proposed attack, a single term $(x^i - x^{N-i})$ can generate $\mathcal{S}^-$ as an $\mathcal{S}$-module, i.e.,

$$(x^i - x^{N-i})(x^j + x^{N-j}) = t_i(x)$$

for some **specific** $i \in \{1, 2, \ldots, (N-1)/2\}$, $j = 1, 2, \ldots, (N-1)/2$, and $t_i(x) \in \mathcal{S}^-$. Thus, we can write:

$$(x^j - x^{N-j}) = (x^i - x^{N-i})d_j(x), \quad \text{for } d_j(x) \in \mathcal{S}. \tag{38}$$

As a result, both h_{s^-} and w'_{s^-} can be expressed as:

$$\begin{aligned} h_{s^-} &= (x^i - x^{N-i})t_0(x) + y(x^i - x^{N-i})t_1(x), \\ w'_{s^-} &= (x^i - x^{N-i})t'_0(x) + y(x^i - x^{N-i})t'_1(x), \end{aligned}$$

for some $t_0(x), t_1(x), t'_0(x), t'_1(x) \in \mathcal{S}$. Since $\theta(h_{s^-}) = \omega'_{s^-}$, we have:

$$\theta(x^i - x^{N-i}) * \underbrace{(t_0(x) + yt_1(x))}_{\in \mathcal{S}} = (x^i - x^{N-i}) * \underbrace{(t'_0(x) + yt'_1(x))}_{\in \mathcal{S}}. \tag{39}$$

Furthermore, since θ maps $\mathcal{S}^-$ to itself, we have:

$$\begin{aligned}\theta(x^i - x^{N-i}) &= \sum_{i=1}^{(N-1)/2} \left(c'_i(x^i - x^{N-i}) + yd'_i(x^i - x^{N-i})\right) \\ &= (x^i - x^{N-i}) * \underbrace{(r_0(x) + yr_1(x))}_{\in \mathcal{S}} \quad \text{(from Equation (38))}.\end{aligned}$$

Substituting this into Equation (39) yields:

$$(x^i - x^{N-i}) * \underbrace{(r_0(x) + yr_1(x))}_{\text{unknown}} * \underbrace{(t_0(x) + yt_1(x))}_{\text{known}} = (x^i - x^{N-i}) * \underbrace{(t'_0(x) + yt'_1(x))}_{\text{known}}. \tag{40}$$

Consequently, one can recover $r(x) = r_0(x) + yr_1(x)$ and determine $\theta(x^i - x^{N-i})$. Then, using Equation (38), we can compute $\theta(x^j - x^{N-j})$ for all $j = 1, 2, \ldots, (N-1)/2$. The final step of the attack involves expressing the ciphertext component e as:

$$\theta(e) = e_0 + \theta(e_1), \qquad \text{for } e_0 \in \mathcal{S} \text{ and } e_1 \in \mathcal{S}^- \tag{41}$$

where reduction of $\theta(e)$ modulo p yields Ψ, which can subsequently be used to recover the message m from E.

Why does Coppersmith's Attack Work? In order for the attacker to succeed, they must find a map θ that cancels the effect of the coefficients introduced by the term $\phi * h * \phi'$, without disturbing the message component ψ. In the construction by Hoffstein and Silverman, although it is difficult to extract ω directly from the public key equation h, computing the expression $h + yhy$ yields $p(\omega + y\omega y)$, which facilitates the construction of an element ω' such that $\theta(h) = p\omega'$ (as discussed in point 2). Obtaining such an element with small coefficients is feasible because both f and F belong to $\mathcal{S}$ (i.e., the *commutative* subring), and are multiplicative inverses of each other. In contrast, if either $f \notin \mathcal{S}$ or $F \notin \mathcal{S}$, or if the product $f * F$ has large coefficients, it becomes difficult to recover a small ω' from the expression $h + yhy$.

E Multiplication of Symmetric and Antisymmetric Elements

For symmetric elements

$$f = (f_{00} + yf_{10}) + \sum_{i=1}^{(N-1)/2} (f_{0i} + yf_{10})(x^i + x^{N-i}),$$

$$g = (g_{00} + yg_{10}) + \sum_{i=1}^{(N-1)/2} (g_{0i} + yg_{10})(x^i + x^{N-i}),$$

and an antisymmetric element

$$h = \sum_{i=1}^{(N-1)/2} (h_{0i} + yh_{10})(x^i - x^{N-i}),$$

– the multiplication between two symmetric elements is given by

$$\begin{aligned} f * g =& (f_{00}g_{00} - f_{10}g_{10} + y(f_{10}g_{00} + f_{00}g_{10})) \\ &+ \sum_{i=1}^{\frac{N-1}{2}} (f_{00}g_{0i} - f_{10}g_{1i} + f_{0i}g_{00} - f_{1i}g_{10} + y(f_{10}g_{1i} + f_{00}g_{1i} + f_{1i}g_{00} + f_{0i}g_{10}))(x^i + x^{N-i}) \\ &+ \sum_{i=1}^{\frac{N-1}{2}} \sum_{j=1}^{\frac{N-1}{2}} (f_{0i}g_{0j} - f_{1i}g_{1j} + y(f_{1i}g_{0j} + f_{0i}g_{1j}))(x^j + x^{N-j}) * (x^i + x^{N-i}) \end{aligned}$$

that costs $(N+1)^2$ integer multiplications.

– the multiplication between a symmetric and an antisymmetric element is given by

$$\begin{aligned} f * h =& \sum_{i=0}^{\frac{N-1}{2}} (f_{00}h_{0i} - f_{10}h_{1i} + y(f_{10}h_{0i} + f_{00}h_{1i}))(x^i - x^{N-i}) \\ &+ \sum_{i=1}^{\frac{N-1}{2}} \sum_{j=1}^{\frac{N-1}{2}} (f_{0j}h_{0i} - f_{1j}h_{1i} + y(f_{0j}h_{1i} + f_{1j}h_{0i}))(x^j + x^{N-j}) * (x^i - x^{N-i}) \end{aligned}$$

that costs $N^2 - 1$ integer multiplications.

F Matrix Multiplication

Let $\mathbf{R}_0, \mathbf{H}$, and $\mathbf{R}_1 \in \mathbb{Z}^{n \times n}$. The product $\mathbf{R}_0\mathbf{H}\mathbf{R}_1$ is given by:

$$\mathbf{R}_0\mathbf{H}\mathbf{R}_1 = \begin{pmatrix} \langle (\mathbf{R}_0\mathbf{H})^{(0,:)}, \mathbf{R}_1^{(:,0)} \rangle & \langle (\mathbf{R}_0\mathbf{H})^{(0,:)}, \mathbf{R}_1^{(:,1)} \rangle & \dots & \langle (\mathbf{R}_0\mathbf{H})^{(0,:)}, \mathbf{R}_1^{(:,n-1)} \rangle \\ \langle (\mathbf{R}_0\mathbf{H})^{(1,:)}, \mathbf{R}_1^{(:,0)} \rangle & \langle (\mathbf{R}_0\mathbf{H})^{(1,:)}, \mathbf{R}_1^{(:,1)} \rangle & \dots & \langle (\mathbf{R}_0\mathbf{H})^{(1,:)}, \mathbf{R}_1^{(:,n-1)} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle (\mathbf{R}_0\mathbf{H})^{(n-1,:)}, \mathbf{R}_1^{(:,0)} \rangle & \langle (\mathbf{R}_0\mathbf{H})^{(n-1,:)}, \mathbf{R}_1^{(:,1)} \rangle & \dots & \langle (\mathbf{R}_0\mathbf{H})^{(n-1,:)}, \mathbf{R}_1^{(:,n-1)} \rangle \end{pmatrix}$$

Here, $(\mathbf{R}_0\mathbf{H})^{(i,:)}$ denotes the i-th row of the matrix product $\mathbf{R}_0\mathbf{H}$, and $\mathbf{R}_1^{(:,j)}$ denotes the j-th column of $\mathbf{R}_1$. Each entry of the resulting matrix is computed as the inner product of a row from $\mathbf{R}_0\mathbf{H}$ and a column from $\mathbf{R}_1$.

In the context of our constructions, we consider the following cases:

(i) $\mathbf{R}_0, \mathbf{R}_1 \in \mathcal{R}_{C_n^+}$, and $\mathbf{H} \in M_n(\mathbb{Z})$, i.e.,

$$\mathbf{R}_0 = \begin{pmatrix} r_{00} & r_{01} & \dots & r_{0(n-1)} \\ -r_{0(n-1)} & r_{00} & \dots & r_{0(n-2)} \\ \vdots & \vdots & \ddots & \vdots \\ -r_{01} & -r_{02} & \dots & r_{00} \end{pmatrix}, \mathbf{R}_1 = \begin{pmatrix} r_{10} & r_{11} & \dots & r_{1(n-1)} \\ -r_{1(n-1)} & r_{10} & \dots & r_{1(n-2)} \\ \vdots & \vdots & \ddots & \vdots \\ -r_{11} & -r_{12} & \dots & r_{10} \end{pmatrix}, \mathbf{H} = \begin{pmatrix} h_0 & h_1 & \dots & h_{n-1} \\ h_n & h_{n+1} & \dots & h_{2n-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n(n-1)} & h_{n(n-1)+1} & \dots & h_{n^2-1} \end{pmatrix}$$

Therefore, the entry $\mathbf{R}_0\mathbf{H}\mathbf{R}_1^{(0,0)}$ has the following expanded form:

$$\begin{aligned}\langle(\mathbf{R}_0\mathbf{H})^{(0,:)}, \mathbf{R}_1^{(:,0)}\rangle &= \sum_{i=0}^{n-1}(r_{0i}h_{in})\, r_{10} - \sum_{j=1}^{n-1}\sum_{i=0}^{n-1}(r_{0i}h_{in+j})\, r_{1(n-j) \bmod n} \\ &= (r_{00}h_0 + r_{01}h_n + \ldots + r_{0(n-1)}h_{n(n-1)})r_{10} \\ &\quad - (r_{00}h_1 + r_{01}h_{n+1} + \ldots + r_{0(n-1)}h_{n(n-1)+1})r_{1(n-1)} \\ &\quad - \ldots \\ &\quad - (r_{00}h_{n-1} + r_{01}h_{2n-1} + \ldots + r_{0(n-1)}h_{n^2-1})r_{11}.\end{aligned}$$

Similarly, other entries of the matrix $\mathbf{R}_0\mathbf{H}\mathbf{R}_1$ can be obtained. Therefore,

$$\overrightarrow{\mathbf{R}_0\mathbf{H}\mathbf{R}_1} = \left(\begin{array}{cccc:cccc:c:cccc}
h_0 & -h_{n-1} & \ldots & -h_1 & h_n & -h_{2n-1} & \ldots & -h_{n+1} & \cdots & h_{n(n-1)} & -h_{n^2-1} & \ldots & -h_{n(n-1)+1} \\
h_1 & h_0 & \ldots & -h_2 & h_{n+1} & h_n & \ldots & -h_{n+2} & \cdots & h_{n(n-1)+1} & h_{n(n-1)} & \ldots & -h_{n(n-1)+2} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
h_{n-1} & h_{n-2} & \ldots & h_0 & h_{2n-1} & h_{2n-2} & \ldots & h_n & \cdots & h_{n^2-1} & h_{n^2-2} & \ldots & h_{n(n-1)} \\ \hdashline
h_n & -h_{2n-1} & \ldots & -h_{n+1} & h_{2n} & -h_{3n-1} & \ldots & -h_{2n+1} & \cdots & -h_0 & h_{n-1} & \ldots & h_1 \\
h_{n+1} & h_n & \ldots & -h_{n+2} & h_{2n+1} & h_{2n} & \ldots & -h_{2n+2} & \cdots & -h_1 & -h_0 & \ldots & h_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
h_{2n-1} & h_{2n-2} & \ldots & h_n & h_{3n-1} & h_{3n-2} & \ldots & h_{2n} & \cdots & -h_{n-1} & -h_{n-2} & \ldots & -h_0 \\ \hdashline
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \hdashline
h_{n(n-1)} & -h_{n^2-1} & \ldots & -h_{n(n-1)+1} & -h_0 & h_{n-1} & \ldots & h_1 & \cdots & -h_{n(n-2)} & h_{n(n-1)-1} & \ldots & h_{n(n-2)+1} \\
h_{n(n-1)+1} & h_{n(n-1)} & \ldots & -h_{n(n-1)+2} & -h_1 & -h_0 & \ldots & h_2 & \cdots & -h_{n(n-2)+1} & -h_{n(n-2)} & \ldots & h_{n(n-2)+2} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
h_{n^2-1} & h_{n^2-2} & \ldots & h_{n(n-1)} & -h_{n-1} & -h_{n-2} & \ldots & -h_0 & \cdots & -h_{n(n-1)-1} & -h_{n(n-1)-2} & \ldots & -h_{n(n-2)}
\end{array}\right)
\left(\begin{array}{c}
r_{00}r_{10} \\ r_{00}r_{11} \\ \vdots \\ \hdashline r_{00}r_{1(n-1)} \\ r_{01}r_{10} \\ r_{01}r_{11} \\ \vdots \\ \hdashline r_{01}r_{1(n-1)} \\ \vdots \\ \hdashline r_{0(n-1)}r_{10} \\ r_{0(n-1)}r_{11} \\ \vdots \\ r_{0(n-1)}r_{1(n-1)}
\end{array}\right)$$

This can be equivalently written using polynomial multiplication over the ring $\mathcal{R}_{C_n^+}$ as:

$$\overrightarrow{\mathbf{R}_0\mathbf{H}\mathbf{R}_1}^{\text{poly}} = \begin{pmatrix} h_0(x) & h_1(x) & \ldots & h_{n-1}(x) \\ h_1(x) & h_2(x) & \ldots & -h_0(x) \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-1}(x) & -h_0(x) & \ldots & -h_{n-2}(x) \end{pmatrix} \begin{pmatrix} r_{00}r_1(x) \\ r_{01}r_1(x) \\ \vdots \\ r_{0(n-1)}r_1(x) \end{pmatrix} \tag{42}$$

where each polynomial $h_i(x)$ is constructed as:

$h_i(x) = h_{in} + h_{in+1}x + \ldots + h_{in+(n-1)}x^{n-1}$ and $r_1(x) = r_{10} + r_{11}x + \ldots + r_{1(n-1)}x^{n-1}$.

For the special case where $\mathbf{H} \in \mathcal{R}_{C_n^+}$, $\overrightarrow{\mathbf{R}_0\mathbf{H}\mathbf{R}_1}^{\text{poly}}$ looks like

$$\overrightarrow{\mathbf{R}_0\mathbf{H}\mathbf{R}_1}^{\text{poly}} = \begin{pmatrix} h_0(x) & x*h_0(x) & \ldots & x^{n-1}*h_0(x) \\ x*h_0(x) & x^2*h_0(x) & \ldots & -h_0(x) \\ \vdots & \vdots & \ddots & \vdots \\ x^{n-1}*h_0(x) & -h_0(x) & \ldots & -x^{n-2}*h_0(x) \end{pmatrix} \begin{pmatrix} r_{00}r_1(x) \\ r_{01}r_1(x) \\ \vdots \\ r_{0(n-1)}r_1(x) \end{pmatrix} = \begin{pmatrix} h_0(x)*r_0(x)*r_1(x) \\ x*h_0(x)*r_0(x)*r_1(x) \\ \vdots \\ x^{n-1}*h_0(x)*r_0(x)*r_1(x) \end{pmatrix}$$

which corresponds to the standard case of NTRU, where the lattice can be constructed from the first row alone, as the remaining rows are merely cyclic rotations of it. Hence, in a ciphertext attack, the message m can be recovered by solving the CVP in a lattice of dimension $2n$ only.

(ii) $\mathbf{R}_0, \mathbf{R}_1 \in \mathcal{R}_{C_{2N}^+}$, and $\mathbf{H} \in \mathcal{R}_{D_N^+}$, i.e.,

$$\mathbf{R}_0 = \left(\begin{array}{ccc:ccc} r_{00} & r_{01} & \dots & r_{0N-1} & r_{0N} & r_{0N+1} & \dots r_{02N-1} \\ -r_{02N-1} & r_{00} & \dots & r_{0N-2} & r_{0N-1} & r_{0N} & \dots r_{02N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \vdots \\ -r_{0N+1} & -r_{0N+2} & \dots & r_{00} & r_{01} & r_{02} & \dots r_{0N} \\ \hdashline -r_{0N} & -r_{0N+1} & \dots & -r_{02N-1} & r_{00} & r_{01} & \dots r_{0N-1} \\ -r_{0N-1} & -r_{0N} & \dots & -r_{02N-2} & -r_{02N-1} & r_{00} & \dots r_{0N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \vdots \\ -r_{01} & -r_{02} & \dots & -r_{0N} & -r_{0N+1} & -r_{0N+2} & \dots r_{00} \end{array}\right), \quad \mathbf{R}_1 = \left(\begin{array}{ccc:ccc} r_{10} & r_{11} & \dots & r_{1N-1} & r_{1N} & r_{1N+1} & \dots r_{12N-1} \\ -r_{12N-1} & r_{10} & \dots & r_{1N-2} & r_{1N-1} & r_{1N} & \dots r_{12N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \vdots \\ -r_{1N+1} & -r_{1N+2} & \dots & r_{10} & r_{11} & r_{12} & \dots r_{1N} \\ \hdashline -r_{1N} & -r_{1N+1} & \dots & -r_{12N-1} & r_{10} & r_{11} & \dots r_{1N-1} \\ -r_{1N-1} & -r_{1N} & \dots & -r_{12N-2} & -r_{12N-1} & r_{10} & \dots r_{1N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \vdots \\ -r_{11} & -r_{12} & \dots & -r_{1N} & -r_{1N+1} & -r_{1N+2} & \dots r_{10} \end{array}\right), \quad \mathbf{H} = \left(\begin{array}{ccc:ccc} h_0 & h_1 & \dots & h_{N-1} & h_N & h_{N+1} & \dots h_{2N-1} \\ h_{N-1} & h_0 & \dots & h_{N-2} & h_{N+1} & h_{N+2} & \dots h_N \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \vdots \\ h_1 & h_2 & \dots & h_0 & h_{2N-1} & h_N & \dots h_{2N-2} \\ \hdashline -h_N & -h_{N+1} & \dots & -h_{2N-1} & h_0 & h_1 & \dots h_{N-1} \\ -h_{N+1} & -h_{N+2} & \dots & -h_N & h_{N-1} & h_0 & \dots h_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \vdots \\ -h_{2N-1} & -h_N & \dots & -h_{2N-2} & h_1 & h_2 & \dots h_0 \end{array}\right).$$

One can observe that the matrix representations of elements in $\mathcal{R}_{C_{2N}^+}$ and $\mathcal{R}_{D_N^+}$ share a common structure of the form: $\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ -\mathbf{B} & \mathbf{A} \end{pmatrix}$. In the case of $\mathcal{R}_{C_{2N}^+}$, the entire matrix is (twisted) right circulant. For $\mathcal{R}_{D_N^+}$, the submatrix $\mathbf{A}$ is right circulant, while $\mathbf{B}$ is left circulant. Moreover, the product $\mathbf{R}_0\mathbf{H}\mathbf{R}_1$ exhibits a similar block pattern, although the resulting submatrices $\mathbf{A}$ and $\mathbf{B}$ may not retain any clearly linkable structure.

Under the semi-structured construction, $\overrightarrow{\mathbf{R}_0\mathbf{H}\mathbf{R}_1}^{\text{poly}}$ can be represented using polynomial multiplications, similar to Equation (42), with the difference that the polynomials $h_i(x), r_0(x) \in \mathcal{R}_{C_{2N}^+}$. Here, the coefficient vector of each polynomial $h_i(x)$ corresponds to the i-th row of the matrix $\mathbf{H}$.

(iii) $r_0 = r'_{00}(x) + yr'_{01}(x), r_1 = r'_{10}(x) + yr'_{11}(x) \in \mathcal{S}$, and $h = h_0(x) + yh_1(x) \in \mathcal{S}^-$, i.e., the corresponding matrices representation is as follows:

$$\mathbf{R}_0 = \left(\begin{array}{ccc:ccc} r_{00} & r_{01} & \dots & r_{01} & r_{0N} & r_{0N+1} & \dots r_{0N+1} \\ r_{01} & r_{00} & \dots & r_{02} & r_{0N+1} & h_{0N+2} & \dots r_{0N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \vdots \\ r_{01} & r_{02} & \dots & r_{00} & r_{0N+1} & r_{0N} & \dots r_{0N+2} \\ \hdashline -r_{0N} & -r_{0N+1} & \dots & -r_{0N+1} & r_{00} & r_{01} & \dots r_{01} \\ -r_{0N+1} & -r_{0N+2} & \dots & -r_{0N} & r_{01} & r_{00} & \dots r_{02} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \vdots \\ -r_{0N+1} & -r_{0N} & \dots & -r_{0N+2} & r_{01} & r_{02} & \dots r_{00} \end{array}\right), \quad \mathbf{R}_1 = \left(\begin{array}{ccc:ccc} r_{10} & r_{11} & \dots & r_{11} & r_{1N} & r_{1N+1} & \dots r_{1N+1} \\ r_{11} & r_{10} & \dots & r_{12} & r_{1N+1} & h_{1N+2} & \dots r_{1N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \vdots \\ r_{11} & r_{12} & \dots & r_{10} & r_{1N+1} & r_{1N} & \dots r_{1N+2} \\ \hdashline -r_{1N} & -r_{1N+1} & \dots & -r_{1N+1} & r_{10} & r_{11} & \dots r_{11} \\ -r_{1N+1} & -r_{1N+2} & \dots & -r_{1N} & r_{11} & r_{10} & \dots r_{12} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \vdots \\ -r_{1N+1} & -r_{1N} & \dots & -r_{1N+2} & r_{11} & r_{12} & \dots r_{10} \end{array}\right), \quad \mathbf{H} = \left(\begin{array}{ccc:ccc} h_{00} & h_{01} & \dots & -h_{01} & h_{10} & h_{11} & \dots -h_{11} \\ -h_{01} & h_{00} & \dots & -h_{02} & h_{11} & h_{12} & \dots h_{10} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \vdots \\ h_{01} & h_{02} & \dots & h_{00} & -h_{11} & h_{10} & \dots -h_{12} \\ \hdashline -h_{10} & -h_{11} & \dots & h_{11} & h_{00} & h_{01} & \dots -h_{01} \\ -h_{11} & -h_{12} & \dots & -h_{10} & -h_1 & h_{00} & \dots -h_{02} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \vdots \\ h_{11} & -h_{10} & \dots & h_{12} & h_{01} & h_{02} & \dots h_{00} \end{array}\right).$$

$\overrightarrow{\mathbf{R}_0\mathbf{H}\mathbf{R}_1}^{\text{poly}}$ for the structured case is given in the "polynomial" form as follows:

$$\overrightarrow{\mathbf{R}_0\mathbf{H}\mathbf{R}_1}^{\text{poly}} = \underbrace{\left(\begin{array}{ccc:ccc} h_0(x)+yh_1(x) & \dots & x^{N-1}(h_0(x)+yh_1(x)) & y(h_0(x)+yh_1(x)) & \dots & yx^{N-1}(h_0(x)+yh_1(x)) \\ x(h_0(x)+yh_1(x)) & \dots & h_0(x)+yh_1(x) & yx^{N-1}(h_0(x)+yh_1(x)) & \dots & yx^{N-2}(h_0(x)+yh_1(x)) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x^{N-1}(h_0(x)+yh_1(x)) & \dots & x^{N-2}(h_0(x)+yh_1(x)) & yx(h_0(x)+yh_1(x)) & \dots & y(h_0(x)+yh_1(x)) \\ \hdashline y(h_0(x)+yh_1(x)) & \dots & yx^{N-2}(h_0(x)+yh_1(x)) & -(h_0(x)+yh_1(x)) & \dots & -x^{N-1}(h_0(x)+yh_1(x)) \\ yx(h_0(x)+yh_1(x)) & \dots & y(h_0(x)+yh_1(x)) & -x^{N-1}(h_0(x)+yh_1(x)) & \dots & -x^{N-2}(h_0(x)+yh_1(x)) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ yx^{N-1}(h_0(x)+yh_1(x)) & \dots & yx^{N-2}(h_0(x)+yh_1(x)) & -x(h_0(x)+yh_1(x)) & \dots & -(h_0(x)+yh_1(x)) \end{array}\right)}_{\mathbf{H}^{\text{poly}}} \left(\begin{array}{c} r_{00}(r'_{10}(x)+yr'_{11}(x)) \\ r_{01}(r'_{10}(x)+yr'_{11}(x)) \\ \vdots \\ r_{01}(r'_{10}(x)+yr'_{11}(x)) \\ r_{0N}(r'_{10}(x)+yr'_{11}(x)) \\ r_{0N+1}(r'_{10}(x)+yr'_{11}(x)) \\ \vdots \\ r_{0N+1}(r'_{10}(x)+yr'_{11}(x)) \end{array}\right). \tag{43}$$

G Selection of Twisted Dihedral Group Ring

The analysis of the underlying algebra in the (twisted) group ring NTRU framework and the structure of its associated lattice is a crucial aspect of secure cryptosystem design. In the following, we provide insights into the structure of the lattice and consequently our reason to work with the twisted dihedral group ring

$$\mathcal{R}_{D_N^+}^q = \mathbb{Z}_q^{\lambda_2} D_N \approx \frac{\mathbb{Z}_q[x,y]}{(x^N - 1, y^2 + 1, xy - yx^{N-1})},$$

where $\lambda_2 : D_N \times D_N \to \mathbb{Z}_q^\times$ is a 2-cocyle defined as

$$\lambda_2(y^{j_1}x^{i_1}, y^{j_2}x^{i_2}) = \begin{cases} -1, & \text{for } i_1, i_2 \in \{0, 1, \ldots, N-1\} \\ & \text{and } j_1 = j_2 = 1 \\ 1, & \text{otherwise.} \end{cases} \tag{44}$$

It should be noted that the matrices associated with the elements of $\mathcal{R}^q_{D_N^+}$ have the following pattern

$$\mathcal{B}^{CS}_{D_N^+} = \begin{pmatrix} \mathbf{I}_N & \mathbf{0}_N & \mathbf{H}_0 & \mathbf{H}_1 \\ \mathbf{0}_N & \mathbf{I}_N & -\mathbf{H}_1 & \mathbf{H}_0 \\ \mathbf{0}_N & \mathbf{0}_N & q\mathbf{I}_N & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{0}_N & \mathbf{0}_N & q\mathbf{I}_N \end{pmatrix}. \tag{45}$$

This structure is similar to the matrix representation of elements of the twisted cyclic group ring

$$\mathcal{R}^q_{C_n^+} = \mathbb{Z}_q^{\lambda_1} C_n \approx \frac{\mathbb{Z}_q[x]}{(x^n+1)},\ n \text{ is even (usually power of 2)},$$

where $\lambda_1 : C_n \times C_n \mapsto \mathbb{Z}_q^\times$ is a 2-cocyle defined as

$$\lambda_1(x^i, x^j) = \begin{cases} 1, & \text{for } i + j < n \\ -1, & \text{for } i + j \geq n \end{cases} \tag{46}$$

The main difference between the two matrix representations is that in the case of $\mathcal{R}^q_{C_n^+}$, the matrix $\begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_1 \\ -\mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix}$ is itself a (twisted) right circulant matrix, while in the case of $\mathcal{R}^q_{D_N^+}$, $\mathbf{H}_0$ is right circulant and $\mathbf{H}_1$ is a left circulant matrix.

The ring $\mathcal{R}^q_{C_n^+}$ has been used in various well-known lattice-based schemes, for example, Falcon [14] and Kyber [44], etc. The benefit of using the ring $\mathcal{R}^q_{C_n^+}$ lies in its algebraic structure that prevents certain attacks, like Gentry's attack [20], due to the fact that "the polynomial $x^n + 1$ does not have any polynomial factor modulus q with small degree, e.g. all the factors of degree $n/2$ have at least Euclidean norm $\sqrt{q}$" [44, Page 35]. On the contrary, for n (a power of 2 or even composite), the polynomial $x^n - 1$ splits into $x^{n/2} - 1$ and $x^{n/2} + 1$, two polynomials of half the degree and small norm $\sqrt{2}$. As a result,

$$\frac{\mathbb{Z}_q[x]}{(x^n-1)} \longrightarrow \frac{\mathbb{Z}_q[x]}{(x^{n/2}-1)} \times \frac{\mathbb{Z}_q[x]}{(x^{n/2}+1)} \quad \text{(Chinese Remainder Theorem)} \tag{47}$$

This allows mapping the secret key $f = f_0(x) + x^{n/2} f_1(x), g(x) = g_0(x) + x^{n/2} g_1(x)$ with coefficients from the set $\{-1, 0, 1\}$ to short polynomials

$$f_0(x)+f_1(x), g_0(x)+g_1(x) \in \frac{\mathbb{Z}_q[x]}{(x^{n/2}-1)},\ f_0(x)-f_1(x), g_0(x)-g_1(x) \in \frac{\mathbb{Z}_q[x]}{(x^{n/2}+1)} \tag{48}$$

with coefficients from the set $\{0, \pm 1, \pm 2\}$, as well as maintaining the relation

$$\begin{aligned} h_0(x) + h_1(x) &= (f_0(x) + f_1(x))^{-1} * (g_0(x) + g_1(x)) \pmod q \\ h_0(x) - h_1(x) &= (f_0(x) - f_1(x))^{-1} * (g_0(x) - g_1(x)) \pmod q. \end{aligned}$$

Therefore, instead of attacking a $2n$-dimensional lattice $\mathcal{L}_{C_n}^{CS}$ generated by the matrix

$$\mathcal{B}_{C_n}^{CS} = \begin{pmatrix} \mathbf{I}_{n/2} & \mathbf{0}_{n/2} & \mathbf{H}_0 & \mathbf{H}_1 \\ \mathbf{0}_{n/2} & \mathbf{I}_{n/2} & \mathbf{H}_1 & \mathbf{H}_0 \\ \mathbf{0}_{n/2} & \mathbf{0}_{n/2} & q\mathbf{I}_{n/2} & \mathbf{0}_{n/2} \\ \mathbf{0}_{n/2} & \mathbf{0}_{n/2} & \mathbf{0}_{n/2} & q\mathbf{I}_{n/2} \end{pmatrix},$$

an attacker can perform lattice attacks on two n-dimensional lattices $\mathcal{L}_{C_n}^{CS,+}$ and $\mathcal{L}_{C_n}^{CS,-}$ generated by the matrices

$$\mathcal{B}_{C_n}^{CS,+} = \begin{pmatrix} \mathbf{I}_{n/2} & \mathbf{H}_0 + \mathbf{H}_1 \\ \mathbf{0}_{n/2} & q\mathbf{I}_{n/2} \end{pmatrix} \text{ and } \mathcal{B}_{C_n}^{CS,-} = \begin{pmatrix} \mathbf{I}_{n/2} & \mathbf{H}_0 - \mathbf{H}_1 \\ \mathbf{0}_{n/2} & q\mathbf{I}_{n/2} \end{pmatrix},$$

respectively, to recover the short vector $(\mathbf{f}_0+\mathbf{f}_1, \mathbf{g}_0+\mathbf{g}_1) \in \mathcal{L}_{C_n}^{CS,+}$ and $(\mathbf{f}_0-\mathbf{f}_1, \mathbf{g}_0-\mathbf{g}_1) \in \mathcal{L}_{C_n}^{CS,-}$, and lifting them back to recover $(\mathbf{f}, \mathbf{g}) \in \mathcal{L}_{C_n}^{CS}$. Since the cost of the lattice attack is directly proportional to the dimension of the lattice, halving the dimension reduces the attack cost drastically. Consequently, an NTRU-like scheme over the ring $\mathbb{Z}[x]/(x^n-1)$ with n as a power of 2 or even a composite number is not considered secure, and it is recommended to select n to be a prime number.

Interestingly, the slight modification of replacing ring $\mathbb{Z}_q[x]/(x^n-1)$ in the NTRU-framework to $\mathbb{Z}_q[x]/(x^n+1)$ mitigates Gentry's attack and allows selecting n to be a power of 2 and an appropriate q to make the ring $\mathbb{Z}_q[x]/(x^n+1)$ NTT-friendly for faster polynomial multiplications.

Raya et al. [40] presented another interpretation of Gentry's attack in terms of matrices and extended it to noncommutative algebraic structure used in [3]. We present a glimpse of their idea and associate it with the concerned twisted dihedral group ring.

For the ring $\mathbb{Z}_q[x]/(x^n-1)$ with n as a power of 2, the matrix representation of elements of the ring has the form $\begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_1 \\ \mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix} \in \mathbb{Z}^{n\times n}$ which can be homomorphically mapped to $\mathbf{H}_0 + \mathbf{H}_1,\ \mathbf{H}_0 - \mathbf{H}_1 \in \mathbb{Z}^{n/2\times n/2}$, matrices of half the dimensions. It is equivalent to applying the maps in Equations (47),(48) in terms of polynomials. This leads to exactly the same lattice attack discussed above. Since the matrix representation of elements of the dihedral group ring $\mathbb{Z}D_N$ is also $\begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_1 \\ \mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix} \in \mathbb{Z}^{2N\times 2N}$ with the difference that $\mathbf{H}_0$ and $\mathbf{H}_1$ are right and left circulant, respectively, the same homomorphisms can be used to reduce the lattice dimension by half as shown in [34]. Moreover, if N is even, then the submatrices further possess special structures

$$\mathbf{H}_0 = \begin{pmatrix} \mathbf{H}_{00} & \mathbf{H}_{01} \\ \mathbf{H}_{01} & \mathbf{H}_{00} \end{pmatrix} \text{ and } \mathbf{H}_1 = \begin{pmatrix} \mathbf{H}_{10} & \mathbf{H}_{11} \\ \mathbf{H}_{11} & \mathbf{H}_{10} \end{pmatrix}, \tag{49}$$

that can be exploited using the same homomorphisms to further reduce the dimension by one-fourth as shown in [32]. Therefore, it was recommended in [32] to select N to be prime while designing a GR-NTRU cryptosystem over the dihedral group ring $\mathbb{Z}D_N$.

On the other hand, the only potential information-preserving homomorphisms for the matrix representation $\mathbf{H} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_1 \\ -\mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix} \in \mathbb{Z}^{n\times n}$ of elements of the ring $\mathbb{Z}_q[x]/(x^n+1)$ are $\mathbf{H} \to \alpha\mathbf{H}_0 + \beta\mathbf{H}_1$ where $(\alpha,\beta) \in \{(1,i),(1,-i)\}$ and $i = \sqrt{-1}$. Therefore, an attacker can construct two n-dimensional lattices $\mathcal{L}^{CS,+}_{C_n^+}$ and $\mathcal{L}^{CS,-}_{C_n^+}$ generated by the matrices

$$\mathcal{B}^{CS,+}_{C_n^+} = \begin{pmatrix} \mathbf{I}_{n/2} & \mathbf{H}_0 + i\mathbf{H}_1 \\ \mathbf{0}_{n/2} & q\mathbf{I}_{n/2} \end{pmatrix} \text{ and } \mathcal{B}^{CS,-}_{C_n^+} = \begin{pmatrix} \mathbf{I}_{n/2} & \mathbf{H}_0 - i\mathbf{H}_1 \\ \mathbf{0}_{n/2} & q\mathbf{I}_{n/2} \end{pmatrix},$$

respectively, to recover the short vector $(\mathbf{f}_0 + i\mathbf{f}_1, \mathbf{g}_0 + i\mathbf{g}_1) \in \mathcal{L}^{CS,+}_{C_n^+}$ and $(\mathbf{f}_0 - i\mathbf{f}_1, \mathbf{g}_0 - i\mathbf{g}_1) \in \mathcal{L}^{CS,-}_{C_n^+}$. However, the existing lattice reduction algorithms are not designed for matrices with complex numbers, and hence, the attacker must map these complex matrices to ones with real entries to perform lattice attacks, thus effectively not gaining any advantage in terms of dimension reduction.

Overall, the structure of matrices of twisted dihedral group ring $\mathcal{R}^q_{D_N^+}$ and the twisted cyclic group ring $\mathcal{R}^q_{C_n^+}$ are similar. Moreover, in our setup, we select N to be prime for the ring $\mathcal{R}^q_{D_N^+}$ so that the submatrices $\mathbf{H}_0$ and $\mathbf{H}_1$ do not have any patterns that expose them to further dimension reduction. While in case of $\mathcal{R}_{C_n^+}$, n is usually chosen to be a power of 2 that results in the following structure of the matrices

$$\mathbf{H}_{C_n^+} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_1 \\ -\mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix} = \begin{pmatrix} \mathbf{H}_{00} & \mathbf{H}_{01} & \mathbf{H}_{10} & \mathbf{H}_{11} \\ -\mathbf{H}_{11} & \mathbf{H}_{00} & \mathbf{H}_{01} & \mathbf{H}_{10} \\ -\mathbf{H}_{10} & -\mathbf{H}_{11} & \mathbf{H}_{00} & \mathbf{H}_{01} \\ -\mathbf{H}_{01} & -\mathbf{H}_{10} & -\mathbf{H}_{11} & \mathbf{H}_{00} \end{pmatrix}. \tag{50}$$

Still, we base the lattice security of our design on the following argument. Any attack on the ring $\mathcal{R}_{D_N^+}$ that reduces the lattice dimension must have an equivalent representation/effect on the matrix $\mathbf{H}_{D_N^+} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_1 \\ -\mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix}$, where $\mathbf{H}_0$ and $\mathbf{H}_1$ possess no further special structure. Due to the similar structure, the same attack must apply to the matrix $\mathbf{H}_{C_n^+} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_1 \\ -\mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix}$ where $\mathbf{H}_0$ and $\mathbf{H}_1$ are further structured.

H Note on Invertibility in Key Generation

The following theorem gives a sufficient condition for the invertibility of symmetric elements in $\mathcal{R}_q$, where q is a power of 2.

Theorem 5. *Let q be a power of 2, N be an odd prime such that 2 is a primitive root modulo N, and $f = a(x) + yb(x) \in \mathcal{L}_f$ be a symmetric element. If $(a_1, b_1) \in \{(0,0),(1,1),(1,-1),(-1,1),(-1,-1)\}$ then f is invertible in $\mathcal{R}_q$.*

Proof. From the discussion in Section 5.3 and the condition (22), it is enough to prove that

$$\det(f)(1) = a(1) * \overline{a(1)} + b(1) * \overline{b(1)} \neq 0 \,(\mathrm{mod}\, 2, \Phi_N(x)). \tag{51}$$

Since f is symmetric, let $a(x) = a_0 + a_1x + a_2x^2 + \ldots + a_2x^{N-2} + a_1x^{N-1}$ and $b(x) = b_0 + b_1x + b_2x^2 + \ldots + b_2x^{N-2} + b_1x^{N-1}$. Then,

$$a(x) \,(\mathrm{mod}\, \Phi_N(x)) = \overline{a(x)} \,(\mathrm{mod}\, \Phi_N(x)) = \underbrace{(a_0 - a_1) + \sum_{i=2}^{\frac{N-1}{2}} (a_i - a_1)(x^i + x^{N-i})}_{=a_\phi(x) \text{ (let)}}$$

$$b(x) \,(\mathrm{mod}\, \Phi_N(x)) = \overline{b(x)} \,(\mathrm{mod}\, \Phi_N(x)) = \underbrace{(b_0 - b_1) + \sum_{i=2}^{\frac{N-1}{2}} (b_i - b_1)(x^i + x^{N-i})}_{=b_\phi(x) \text{ (let)}}.$$

Therefore,

$$\begin{aligned}\det(f)(1) \,(\mathrm{mod} \Phi_N(x)) &= a_\phi(1)^2 + b_\phi(1)^2 \\ &= (a_0 + 2X - (N-2)a_1)^2 + (b_0 + 2Y - (N-2)b_1)^2,\end{aligned}$$

where $X = \sum_{i=2}^{\frac{N-1}{2}} a_i, Y = \sum_{i=2}^{\frac{N-1}{2}} b_i$. Since $f \in \mathcal{T}(d_f+1, d_f)$, therefore, $a_0^2 + b_0^2 = 1$ and

$$\det(f)(1) = 1 + (N-2)^2(a_1^2 + b_1)^2 \neq 0 \,(\mathrm{mod}\, 2, \Phi_N(x)),$$

as $(a_1, b_1) \in \{(0,0), (1,1), (1,-1), (-1,1), (-1,-1)\}$, and N is an odd prime. Therefore, f is invertible in $\mathcal{R}_q \,(\mathrm{mod}\, \Phi_N(x))$. □

Although the restriction on coefficients from Theorem 5 does not guarantee that $\det(f)(1) \neq 0 \,(\mathrm{mod} 3, \Phi_N(x))$, empirical results show that for any randomly sampled $f \in \mathcal{L}_f$, it is typically the case that $\det(f) \neq 0 \,(\mathrm{mod}\, 3, \Phi_N(x))$. This is because, for the determinant to be zero, all its coefficients would need to be multiples of 3, which is almost a negligible occurrence. Nevertheless, one can ensure deterministically that $\det(f) \neq 0 \,(\mathrm{mod} 3, \Phi_N(x))$ by selecting $f_0 \in \mathcal{S} \cap \mathcal{T}(d'_f + 1, d'_f)$ and $f_1 \in \mathcal{S} \cap \mathcal{T}(d'_f, d'_f)$, where $d'_f = d_f/2$, and $a_1 = b_1 = 0$. However, this approach significantly restricts the key space and may introduce unforeseen vulnerabilities. Therefore, we avoid this restriction in our implementation. Instead, we simply verify condition (22) and reject any sample that fails to satisfy it.

I Scope for Future Analysis

NTT for Faster Multiplications One multiplication in the ring $\mathbb{Z}_q^{\lambda_2} D_N$ requires four multiplications in the ring $\mathbb{Z}_q C_N$. Therefore, the cost of schoolbook multiplication in both $\mathbb{Z}_q^{\lambda_2} D_N$ and $\mathbb{Z}_q^{\lambda_1} C_n$, with $n = 2N'$ where N' is the nearest power of two to the prime N, is approximately $4N^2$ integer multiplications.

In the ring $\mathbb{Z}_q^{\lambda_1} C_n$, as n is a power of 2, one can exploit faster polynomial multiplication techniques, such as the Number Theoretic Transform (NTT), by selecting an appropriate modulus q. This is the standard approach adopted in most lattice-based schemes constructed over this ring. In contrast, for $\mathbb{Z}_q^{\lambda_2} D_N$, where N is prime, one cannot take advantage of such fast methods. This raises an interesting question: can we minimally modify the algebraic structure of $\mathbb{Z}_q^{\lambda_2} D_N$ so that it becomes compatible with NTT?

One possible approach is to further twist the multiplication in $\mathbb{Z}_q^{\lambda_2} D_N$ such that the four multiplications in the rings

$$\mathbb{Z}_q C_N \approx \mathbb{Z}_q[x]/(x^N - 1)$$

can instead be interpreted as multiplications in the twisted ring

$$\mathbb{Z}_q^{\lambda_1} C_N \approx \mathbb{Z}_q[x]/(x^N + 1).$$

This can be achieved by defining the following 2-cocycle $\lambda_3 : D_N \times D_N \mapsto \mathbb{Z}_q^\times$,

$$\lambda_3(x^i, x^j) = \begin{cases} 1, & \text{for } i+j < n \\ -1, & \text{for } i+j \geq n \end{cases} \qquad \lambda_3(yx^i, x^j) = \begin{cases} 1, & \text{for } i+j < n \\ -1, & \text{for } i+j \geq n \end{cases}$$

$$\lambda_3(x^i, yx^j) = \begin{cases} 1, & \text{for } j < i \\ -1, & \text{for } j \geq i \end{cases} \qquad \lambda_3(yx^i, yx^j) = \begin{cases} 1, & \text{for } j \geq i \\ -1, & \text{for } j < i \end{cases}.$$

The corresponding twisted dihedral group ring is then given by

$$\mathbb{Z}_q^{\lambda_3} D_N \approx \frac{\mathbb{Z}_q[x, y]}{(x^N + 1, y^2 + 1, xy - yx^{N-1}).} \tag{52}$$

By selecting N to be a power of two and choosing an appropriate q, one can then apply NTT to achieve faster polynomial multiplications in the ring $\mathbb{Z}_q[x]/(x^N + 1)$.

This modification raises important concerns regarding security against lattice attacks, as it alters the algebraic structure and consequently the basis matrix representation. Specifically, two intertwined structures arise in this new twisted group ring setup. The first stems from the dihedral group structure twisted by the relation $y^2 = -1$, leading to the matrix representation

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_1 \\ -\mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix}$$

where $\mathbf{H}_0$ and $\mathbf{H}_1$ are right and left circulant matrices, respectively. The second arises from the cyclic group structure twisted by the relation $x^N = -1$, which is reflected in the block structure of the submatrices

$$\mathbf{H}_0 = \begin{pmatrix} \mathbf{H}_{00} & \mathbf{H}_{01} \\ -\mathbf{H}_{01} & \mathbf{H}_{00} \end{pmatrix} \text{ and } \mathbf{H}_0 = \begin{pmatrix} \mathbf{H}_{10} & \mathbf{H}_{11} \\ -\mathbf{H}_{11} & \mathbf{H}_{10} \end{pmatrix}.$$

Consequently, the matrix representation of elements in the ring $\mathbb{Z}_q^{\lambda_3} D_N$ takes the form

$$\mathbf{H}_{D_N^+}^{\lambda_3} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_1 \\ -\mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix} = \begin{pmatrix} \mathbf{H}_{00} & \mathbf{H}_{01} & \mathbf{H}_{10} & \mathbf{H}_{11} \\ -\mathbf{H}_{01} & \mathbf{H}_{00} & -\mathbf{H}_{11} & \mathbf{H}_{10} \\ -\mathbf{H}_{10} & -\mathbf{H}_{11} & \mathbf{H}_{00} & \mathbf{H}_{01} \\ \mathbf{H}_{11} & -\mathbf{H}_{10} & -\mathbf{H}_{01} & \mathbf{H}_{00} \end{pmatrix}. \tag{53}$$

As discussed, there are no straightforward homomorphisms that reduce the dimension of matrices of the form $\begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_1 \\ -\mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix}$ by mapping them to smaller-sized real matrices. Thus, the outer structure of the twisted dihedral group ring and the inner substructure of the twisted cyclic group ring cannot be independently utilized to reduce lattice dimensions. However, it is not immediately clear how the interaction between the two structures could be exploited to find dimension-reducing homomorphisms. Therefore, a deeper and more rigorous analysis is required to investigate whether any such homomorphisms exist that could take advantage of the new twisted dihedral structure in Equation (53) to reduce lattice dimensions, and further, to understand the implications such mappings might have on the twisted cyclic structure described in Equation (50). This question is therefore left as an open problem, and we highlight it as an important direction for future research.

Digital Signatures and Challenges This work introduces a novel KEM design within the NTRU framework that leverages noncommutative algebra to mitigate direct lattice-based attacks. Beyond key encapsulation, it is also natural to investigate whether this approach can be extended to construct a digital signature scheme. In particular, one could envision a noncommutative analog of Falcon [14], potentially offering stronger resistance against classical lattice attacks.

However, designing such a Falcon-like scheme over the noncommutative algebra of twisted dihedral group rings presents significant challenges. The main difficulty lies in the trapdoor generation process, which involves creating a short basis for the NTRU lattice that facilitates efficient signing. Specifically, given a short secret key $(f, g) \in \mathcal{R}_{C_n^+}$, the task is to find another pair of short polynomials $(F, G) \in \mathcal{R}_{C_n^+}$ such that

$$f * G - g * F = q.$$

Then, $\{(f, g), (F, G)\}$ is also another basis for the NTRU lattice generated by the public basis $\{(1, h), (0, q)\}$ where $h = f^{-1} * g \pmod q$ is the public key.

In the commutative setting $\mathcal{R}_{C_n^+}$, several algorithms have been proposed in the literature for completing an NTRU basis for trapdoor generation. The original method was introduced in [21], which is an initial work on NTRU signatures, and subsequent refinements have yielded more efficient approaches, such as the trapdoor generation algorithms used in Falcon [14, Algorithms 5 and 6].

To our knowledge, however, no analogous method is known for noncommutative dihedral group rings. Formally, given short elements $(f, g) \in \mathcal{R}_{D_N}$ or $\mathcal{R}_{D_N^+}$, finding another pair $(F, G) \in \mathcal{R}_{D_N}$ or $\mathcal{R}_{D_N^+}$ of short elements such that $f * G - g * F = q$ remains an open problem. A solution to this problem would lead to a noncommutative analog of Falcon. Moreover, the problem itself is of independent interest from a mathematical perspective.

Smaller Modulus q The integration of PQC algorithms with the real-world protocols demands a lower communication cost, primarily determined by the

bandwidth. Therefore, it is worth exploring the ways to reduce the value of modulus q, which directly affects the bandwidth.

Some recent works proposed NTRU-like schemes with approaches to achieve a lower modulus. For example, BAT [15] by Fouque et al. altered the encryption to

$$c = hm + e \,(\mathrm{mod} q) \in \mathcal{R}^q_{C_n^+},\ n\text{: power of 2}$$

removing the mask modulus p, thereby requiring smaller values of q for decryption as

$$\|f * c \,(\mathrm{mod} q) = g * r + f * m\|_\infty < \|pg * r + f * m\|_\infty.$$

However, this would require an NTRU trapdoor basis for decoding to retrieve the message. Therefore, the realization of this technique in the noncommutative setting requires a solution to the problem posed above.

Another work, DAWN [36] by Liu et al., employs a double decoding paradigm as

$$c = h * r + e + t^{-1} * w * m \,(\mathrm{mod} q) \in \mathcal{R}^q_{C_n^+},$$

where t, w are masking polynomials with small norm such that $x^n+1 = 0 \pmod{t, p}$ and w is suitably chosen to correct larger errors. Standard NTRU is a particular instantiation of DAWN with $p = 3, t = 1, w = 1$. DAWN proposes to select $p = 2, t = x^{n/2}+1$, and $w = x^{n/4}+1$ so that the successful decryption condition that requires

$$\lfloor t * (g * r + f * e) + f * w * m \rceil_q$$

to be correctly decoded using w, is more relaxed and favorable than the original NTRU condition, resulting in a lower value of modulus q. For these parameters, a simple decoding algorithm [36, Algorithm 2] is provided to recover the message.

Adapting this approach to a noncommutative setup requires much groundwork, including the selection of optimal parameters $t, w \in \mathcal{R}^q_{D_N^+}$ for a relaxed successful decryption condition and developing a similar efficient decoding algorithm for message retrieval.

We leave it as future work to investigate whether these techniques can be incorporated into our framework. Such extensions would require substantial analysis, and it is not yet clear whether they can be seamlessly integrated with our scheme.