An Efficient Noncommutative NTRU from Semidirect Product

Vikas Kumar[1], Ali Raya[2], Aditi Kar Gangopadhyay[1], Sugata Gangopadhyay[2], and Md Tarique Hussain[3]

[1] Department of Mathematics
Indian Institute of Technology Roorkee, Roorkee 247667, India.
{v_kumar, aditi.gangopadhyay}@ma.iitr.ac.in

[2] Department of Computer Science and Engineering
Indian Institute of Technology Roorkee, Roorkee 247667, India.
{ali_r, sugata.gangopadhyay}@cs.iitr.ac.in

[3] Department of Information Technology, Indian Institute of Engineering Science and Technology Shibpur, Howrah 711103, India.
mdtariqueh007@gmail.com

Abstract. NTRU is one of the most extensively studied lattice-based schemes. Its flexible design has inspired different proposals constructed over different rings, with some aiming to enhance security and others focusing on improving performance. The literature has introduced a line of noncommutative NTRU-like designs that claim to offer greater resistance to existing attacks. However, most of these proposals are either theoretical or fall short in terms of time and memory requirements when compared to standard NTRU. To our knowledge, DiTRU (Africacrypt 2024) is the first noncommutative analog of NTRU provided as a complete package. Although DiTRU is practical, it operates at two times slower than NTRU with no decryption failure. Additionally, key generation, encryption, and decryption are 1.2, 1.7, and 1.7 times slower, respectively, with negligible decryption failure. In this work, we introduce a noncommutative version of NTRU that offers comparable performance and key sizes to NTRU while improving upon DiTRU. Our cryptosystem is based on the GR-NTRU framework, utilizing the group ring of a semidirect product of cyclic groups over the ring of Eisenstein integers. This design allows for an efficient construction with key generation speeds approximately two (three) times faster than NTRU (DiTRU). Further, the proposed scheme provides roughly a speed-up by a factor of 1.2 (2) while encrypting/decrypting messages of the same length over NTRU (DiTRU). We provide a reference implementation in C for the proposed cryptosystem to prove our claims.

Keywords: NTRU· GR-NTRU· Semidirect product· Group rings· Eisenstein integers

1 Introduction

NTRU [19], first introduced by Hoffstein, Pipher, and Silverman, is one of the most prominent and efficient lattice-based postquantum cryptosystems. The long cryptanalytic history and absence of effective attacks against well-defined parameter sets of NTRU place it at a strong position in the pyramid of postquantum schemes. The trust in NTRU is further deepened as three NTRU-style schemes [7, 9, 25] reached the third round of NIST's postquantum standardization process. The flexibility in the design of NTRU has resulted in many variants aimed at improving the efficiency and security of NTRU. We will refer to the NTRU version presented in [18] as *standard* NTRU.

Crypanalytic landscape. Broadly, the security of NTRU is based on the NTRU hard assumption, formulated as:

Given a public key h, computed as $h = f^{-1} * g \pmod q$, where f, g are short private polynomials and q is a modulus, find f', g' with small coefficients such that $f' * h = g' \pmod q$.

The most straightforward way to attack the problem is to search for small elements from the underlying ring satisfying the NTRU key equation. One can optimize the search process by incorporating approaches like Meet in the middle attack [20]. The NTRU problem can also be solved by finding short vectors in lattices of particular structures [13] using lattice reduction algorithms [12,32]. In this sense, NTRU is classified as a lattice-based cryptosystem. The two previous approaches can be further combined, resulting in a hybrid attack [21]. Other attacks against NTRU exploit the selected parameters, like decryption failure attacks [22] and subfield attacks [14]. Hence, to propose a set of parameters that target a certain level of security, one needs to consider the cost of all previous attacks. However, in some other scenarios, the attacker may have access to extra information about the cryptosystem that enables different cryptanalysis tools. For example, the NTRU learning problem, phrased as:

Given NTRU public keys $h_i = f^{-1} * g_i \pmod q$, for a fixed f and a number of independently sampled g_i, find f,

was believed to be hard until recently, Kim and Lee [27] introduced a polynomial-time attack that can break it if the attacker has access to n different samples of h_i (where n refers to the extension degree of the NTRU ring $\mathbb{Z}[x]/(x^n - 1)$). A simple analysis of the Kim and Lee attack shows that their method works when the underlying ring is commutative since building the system of equations that leads to attacking the NTRU learning problem is possible only if the attacker can reformulate the equations using commutativity. We refer the reader to the original work [27] for the attack details. Therefore, employing noncommutative algebras to generalize NTRU appears to be a promising research direction. Furthermore, Coppersmith and Shamir [13], in the initial work of lattice attack on NTRU, also hinted that noncommutative structure might prevent their attack and other possible attacks that take benefit of the commutative structure.

Related works. Although several proposals exist for noncommutative NTRU-like cryptosystems, many of them do not maintain the hard assumption of NTRU. The first noncommutative variant of NTRU by Hoffstein and Silverman is an example under this category, where the scheme was vulnerable to attack, which does not apply to standard NTRU. For details of this attack, we refer the readers to [40]. Other proposals uphold the general assumption of NTRU but fall behind in terms of the efficiency and compactness of the parameter sets compared to standard NTRU. For example, QTRU [34], SQTRU [39], OTRU [33] based on quaternion, split quaternion, and octonion algebras, respectively, are 4, 4, and 16 times slower than NTRU for the same level of security. BQTRU's [5] security analysis raises concerns as the authors discuss their parameter selection, conjecturing that Gentry's attack [17] does not challenge the security of their scheme without a rigorous analysis. Further, none of the above constructions provided a full implementation, keeping it unclear how efficiently one can address some of the design aspects, like inverting elements in the new setting of the noncommutative ring.

To our knowledge, DiTRU [35] is the only noncommutative NTRU-like design provided with a full-package implementation. DiTRU is structured as a group ring NTRU over the dihedral group of order $2N$. The hard assumption of NTRU is maintained as the key recovery attack is equal to finding 'short' elements from the underlying noncommutative ring. However, according to the authors, the associated lattice with DiTRU is susceptible to a one-layer Gentry attack, which can reduce the dimension of lattice attacks from $4N$ to $2N$. Consequently, the parameters chosen for DiTRU are twice as large as those used for NTRU to achieve equivalent levels of security without allowing decryption failure. This ratio can be scaled down slightly when a negligible decryption failure is deemed acceptable. In summary, while DiTRU offers a practical noncommutative analog to NTRU, it fails to maintain NTRU performance for equivalent parameter sets.

Our contribution. We design a noncommutative NTRU variant in the GR-NTRU framework [42]. Although GR-NTRU is usually designed over the group rings $\mathbb{Z}G$. To achieve faster multiplication, we make minor modifications and build it over the group ring RG where R is the ring of Eisenstein integers as in ETRU [26]. The group $G = C_N \rtimes C_3$ is the noncommutative semidirect product of cyclic groups C_N and C_3 of order N and 3, respectively. For our construction, we clear all the implementation details and consider the following points:

- **Inversion algorithm:** We provide an inversion algorithm (Algorithm 2) to find invertible elements in the underlying group ring. This algorithm constitutes an essential part of the key generation process. The proposed algorithm introduces a way to check/find invertible elements by mapping the units over the proposed ring $R(C_N \rtimes C_3)$ to the ring RC_N where R is the ring of Eisenstein integers. We provide the constant-time implementation for our algorithm following the Bernstein-Yang algorithm [8]. Our findings demonstrate that the proposed key generation process is faster than the key generation processes for NTRU and DiTRU by a factor of 2 and 3, respectively.
- **Analysis of lattice security:** We give a detailed cryptanalysis of the security of the associated lattices with our construction and analyze the hardness of retrieving the decryption key using the lattice reduction algorithms.

- **Concrete parameter selections:** We model the decryption failure with respect to the chosen design, and accordingly, we provide two sets of parameters: one with zero decryption failure rate, and the other allows a negligible decryption failure. These parameters have been selected considering the best combinatorial and lattice-based attacks against our construction.
- **Reference implementation:** We provide a C reference implementation to prove the claimed results on the performance and compactness of our proposal. Table 4 compares the performance of our construction vs. NTRU and DiTRU while encrypting/decrypting messages of the same length. Our cryptosystem demonstrates improvement over NTRU and DiTRU by a factor of 1.2 and 2, respectively. The implementation is available and can be accessed at `https://github.com/The-Isogeniest/Ei_TRU`.

1.1 Paper Layout

Section 2 contains the required notations and preliminaries. The proposed cryptosystem is given in Section 3. Section 4 gives an analysis of different attacks on the new design. Finally, the cryptosystem's parameters, its performance analysis, and comparison with NTRU and DiTRU are provided in Section 5.

2 Notation and preliminaries

$\mathbb{C}, \mathbb{R}$, and $\mathbb{Z}$ denote the set of complex numbers, real numbers and integers, respectively. Symbol $*$, wherever it occurs, denotes the multiplication of two elements with respect to the underlying algebraic structure, which should be clear from the context. For a positive integer n, $\mathbb{Z}/n\mathbb{Z}$ is the ring of integers modulo n. R denotes a commutative ring with unity and R^n is cartesian product of n copies of R. The norm of a vector $u = (u_1, u_2, \ldots, u_n) \in \mathbb{R}^n$ is defined as $\|u\| = \sqrt{\sum_{i=1}^{n} u_i^2}$. The length/norm of a complex number $\xi = a + \iota b$ is $|\xi| = \sqrt{a^2 + b^2}$, where $\iota = \sqrt{-1} \in \mathbb{C}$ is the imaginary root of unity. Let Re and Im denote the real and imaginary parts of a complex number, respectively. We denote the primitive cube root of unity by ω, i.e., $\omega = e^{\frac{2\pi i}{3}}$, $\omega^3 = 1$ and $\omega \neq 1$. U_n denote the set of nth roots of unity. $U_3 = \{1, \omega, \omega^2 = -1 - \omega\}$ and $U_6 = \{\pm 1, \pm\omega, \pm\omega^2\}$. $M_n(R)$ denotes the ring of $n \times n$ matrices with entries from the ring R. Sampling an element s uniformly at random from a set S is denoted by $s \xleftarrow{\$} S$. We may define more notations in the course of the paper, wherever required.

2.1 Lattices

Definition 1 (Lattice). *Let $B \in \mathbb{R}^{n \times m}$ with linearly independent rows b_i, for $i = 1, 2, \ldots, n$. A lattice L_B generated by the matrix B is the set of integer linear combination of rows of B, i.e.,*

$$L_B = \left\{ \sum_{i=1}^{n} \gamma_i b_i : \gamma_i \in \mathbb{Z} \right\}. \tag{1}$$

The matrix B is called a basis matrix of the lattice L_B. The determinant of the lattice L_B is given by $\sqrt{det(B^T B)}$ and is independent of the choice of basis. If all the rows of B have integer entries, we say that the lattice is an integral lattice. This paper deals with only full-rank integral lattices, i.e., $n = m$. A full rank lattice $L_B \subset \mathbb{Z}^n$ is called q-ary lattice for some $q > 0$, if $q\mathbb{Z}^n \subset L_B \subset \mathbb{Z}^n$.

Definition 2 (SVP). *The Shortest Vector Problem (SVP) is to find a non-zero vector $u \in L_B$ such that*

$$\|u\| = \min_{w \in L_B - \{0\}} \|w\|.$$

We denote the length of the shortest vector in lattice L_B by $\lambda_1(L_B)$.

Definition 3 (CVP). *Closest Vector Problem (CVP) is to find a vector $v \in L_B$ closest to the given target vector $t \in \mathbb{R}^d$, i.e., $||v - t|| \leq ||w - t||$ for all $w \in L_B$.*

Definition 4 (Gaussian heuristic). *Suppose $L_B \subset \mathbb{R}^n$ is a lattice generated by matrix $B \in \mathbb{R}^{n \times n}$. Gaussian heuristic estimates the length of the shortest vector in the lattice L_B to be*

$$\sigma(L_B) = \sqrt{n/2\pi e} \cdot \ det(B)^{1/n}. \tag{2}$$

2.2 Semidirect product of cyclic groups

Definition 5. *[16, Definition 2.2] Given two groups G and H and a group homomorphism $\phi : H \to Aut(G)$ (the automorphism group of G), the Semidirect Product of G and H with respect to ϕ, denoted $G \rtimes_\phi H$(or, simply, $G \rtimes H$) is a new group with set $G \times H$ and multiplication operation $(g_1, h_1)(g_2, h_2) = (g_1\phi(h_1)(g_2), h_1h_2)$.*

The fact that $Aut(C_N) \cong \mathbb{Z}_{N-1}$ gives the following result:

Theorem 1. *[16, Proposition 2.1] Let $C_N \cong \frac{\mathbb{Z}}{N\mathbb{Z}}$ and $C_M \cong \frac{\mathbb{Z}}{M\mathbb{Z}}$ be two cyclic groups of order N and M, respectively. A semidirect product $C_N \rtimes_k C_M$ corresponds to a choice of integer k such that $k^M \equiv 1 \bmod N$. The semidirect product group is given by*

$$C_N \rtimes_k C_M = \left\langle x, y \mid x^N = y^M = 1, yxy^{-1} = x^k \right\rangle. \tag{3}$$

When there is no confusion of k, we denote the semidirect product by $C_N \rtimes C_M$.

In our work, we consider the case when N is prime, $M = 3$, and $3|(N-1)$ so that we have a noncommutative semidirect product $C_N \rtimes_k C_3$ for some $k \not\equiv 1 \bmod N$ such that $k^3 \equiv 1 \bmod N$. Let us fix such a k and order the elements of the group $C_N \rtimes_k C_3$ as follows:

$$C_N \rtimes_k C_3 = \{1, x, \ldots, x^{N-1}, y, yx, \ldots, yx^{N-1}, y^2, y^2x, \ldots, y^2x^{N-1}\}.$$

Theorem 2. *[15, Section 5.5] Let H be a finite cyclic group and N be an arbitrary group. Suppose $\phi_1, \phi_2 : H \to Aut(N)$ ($Aut(N)$ is the group of automorphisms on N) are homomorphisms such that $Im(\phi_1)$ and $Im(\phi_2)$ are conjugate subgroups of $Aut(N)$. Then $N \rtimes_{\phi_1} H \cong N \rtimes_{\phi_2} H$.*

Corollary 1. *Let N be a prime number such that $3|(N-1)$, then there exists only one noncommutative semidirect product $C_N \rtimes_k C_3$ unique up to isomorphism.*

Proof. Since N is prime, therefore $Aut(C_N) \cong C_{N-1}$ (cyclic group of order $N-1$). Hence, there is one and only one subgroup of order 3 of $Aut(C_N)$ because $3|(N-1)$. Consequently, for any two non-trivial homomorphisms $\phi_1, \phi_2 : C_3 \to Aut(C_N)$, we have $Im(\phi_1) = Im(\phi_2)$. Thus, $C_N \rtimes_{\phi_1} C_3 \cong C_N \rtimes_{\phi_2} C_3$. □

2.3 Ring of Eisenstein integers

We briefly discuss the essential properties of Eisenstein integers that are given in [26] in detail. The ring of Eisenstein integers is defined as

$$\mathbb{Z}[\omega] = \{a + b\omega : a, b \in \mathbb{Z}\} = \left\{ a - \frac{b}{2} + i\frac{b\sqrt{3}}{2} : a, b \in \mathbb{Z} \right\}. \tag{4}$$

The length of an Eisenstein integer $z = a+b\omega$ is $|z| = \sqrt{a^2 + b^2 - ab}$. The product of two Eisenstein integers is given by

$$(a + b\omega) * (c + d\omega) = ac - bd + (ac + (a - b)(d - c))\omega. \tag{5}$$

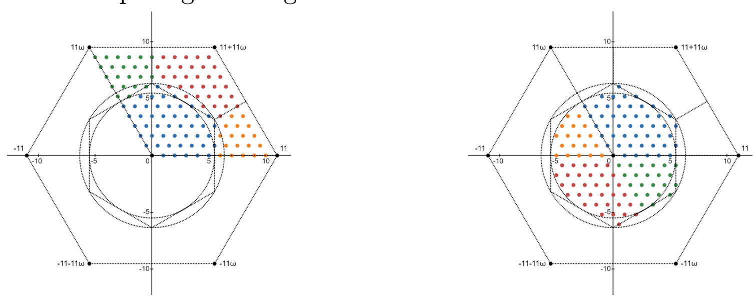
Therefore, one product in $\mathbb{Z}[\omega]$ requires 3 multiplications and 4 additions over $\mathbb{Z}$. The map $\langle \cdot \rangle : \mathbb{Z}[\omega] \to M_2(\mathbb{Z})$ given by

$$\langle z \rangle = \begin{pmatrix} a & b \\ -b & a - b \end{pmatrix} \tag{6}$$

is a ring homomorphism and the map $a+b\omega \to (a, b) \in \mathbb{Z}^2$ is an isomorphism. The multiplication $(a + b\omega) * (c + d\omega) \in \mathbb{Z}[\omega]$ can be realized as $(a, b) \cdot \langle c + d\omega \rangle \in \mathbb{Z}^2$. The Voronoi cell V_q of an element $q \in \mathbb{Z}[\omega]$ is the region bounded by a certain regular hexagon inscribed between circles of radius $|q|/2$ and $|q|/\sqrt{3}$ as shown in Figure 1.

Theorem 3. *[26, Theorem 1] The set U_6 consists of exactly all units (invertible elements) of $\mathbb{Z}[\omega]$. The primes of $\mathbb{Z}[\omega]$ are (up to multiplication by a unit): $1-\omega$; rational primes $p \in \mathbb{Z}$ satisfying $p \equiv 2 \pmod 3$; and those $q \in \mathbb{Z}[\omega]$ for which $|q|^2 = p$ is a rational prime satisfying $p \equiv 1 \pmod 3$.*

Division in $\mathbb{Z}[\omega]$. For any α and a nonzero q in $\mathbb{Z}[\omega]$, we say that $\beta \in \mathbb{Z}[\omega]$ is residue or reduced element modulo q corresponding to α, i.e., $\alpha \pmod q = \beta$, if we can write $\alpha = rq + \beta$ where $r \in \mathbb{Z}[\omega]$ is the closest element to $q^{-1}\alpha \in \mathbb{C}$, or equivalently $rq \in \mathbb{Z}[\omega]$ is the nearest multiple of q to α. The set of residues/reduced elements modulo q is denoted as D_q. It should be observed that $\mathbb{Z}[\omega]$ is a regular hexagonal lattice in $\mathbb{C} \cong \mathbb{R}^2$ with basis $\{1, \omega\}$ over $\mathbb{Z}$, and the ideal $\langle q \rangle$ is again a lattice with basis $\{q, q\omega\}$. Therefore, finding $r \in \mathbb{Z}[\omega]$ closest to $q^{-1}\alpha \in \mathbb{C}$ is equivalent to solving Closest Vector Problem (CVP) in the lattice $\mathbb{Z}[\omega]$. A division algorithm over $\mathbb{Z}[\omega]$ is discussed in [26, Algorithm 1] that costs 27 integer multiplications and 32 integer additions, which is significantly costlier than computing an integer modulus.



(a) P_{11} (b) D_{11}

Fig. 1: Division in $\mathbb{Z}[\omega]$ by $q = 11$. The different colors in P_{11} represent regions close to different multiples of $q = 11$, which are possibly $0, 11, 11\omega$, or $11 + 11\omega$. Each colored region in D_{11} represents the residues of elements in its corresponding part in P_{11} translated according to Algorithm 1 depending on their closeness to multiples of 11.

In this work, we propose a more efficient division algorithm (Algorithm 1) that works for division by elements of the form $q + 0\omega$, with a cost of 4 integer multiplications and 4 integer additions. We briefly explain the working of Algorithm 1. Let $a, b \in \mathbb{Z}$ then there exist unique integers r, s and $0 \leq x, y < q$ such that $a = rq + x,\ b = sq + y$. Therefore, $a + b\omega = q(r + s\omega) + (x + y\omega) \equiv x + y\omega \pmod{q}$. Let $P_q = \{x + y\omega : 0 \leq x, y < q\}$, then it is enough to find residues of elements in P_q modulo q. For an element $x + y\omega \in P_q$, Algorithm 1 returns the residue modulo q by locating the nearest multiple of q in $\mathbb{Z}[\omega]$ as follows: If the nearest multiple of q is $x_1 + y_1\omega$, where $x_1, y_1 \in \{0, q\}$, then the residue is $(x - x_1) + (y - y_1)\omega$. When a point is equidistant from two multiples of q, then the algorithm chooses the one on the left. We have shown the regions in P_q closer to different multiples of q in Figure 1a, and the corresponding residues D_q in Figure 1b, for $q = 11$.

Algorithm 1: Division by integers in $\mathbb{Z}[\omega]$

Input: $\alpha = a + b\omega \in \mathbb{Z}[\omega]$, and an element $q = q + 0\omega \in \mathbb{Z}[\omega]$.
Output: $\beta \in \mathbb{Z}[\omega]$ such that $\alpha = rq + \beta$ where $r \in \mathbb{Z}[\omega]$ is nearest to $q^{-1}\alpha$.

$x = a \pmod{q},\ y = b \pmod{q},\ X = 2x,\ Y = 2y$
if $x + y > q,\ X > y,\ Y \geq x$ **then return** $\beta = (x - q) + (y - q)\omega$
if $X - y > q,\ Y < x$ **then return** $\beta = (x - q) + y\omega$
if $Y - x \geq q,\ X \leq y$ **then return** $\beta = x + (y - q)\omega$
else return $\beta = x + y\omega$

Remark 1. Algorithm 1 returns the set of residues D_q modulo q that is almost symmetrically distributed around 0, which is needed to decrease the decryption failure. However, $q = 2$ is a special case where we get $D_2 = \{0, 1, -\omega, -\omega^2\}$ which is not distributed around 0. Observing that $\omega, \omega^2 \equiv -\omega, -\omega^2 \pmod 2$ and $|\pm\omega| = |\pm\omega^2| = 1$. We redefine D_2 as $D_2 = \{0, 1, \omega, \omega^2\}$ by mapping $-\omega \to \omega$ and $-\omega^2 \to \omega^2$. Conclusively, $a + b\omega \pmod 2 = a \pmod 2 + b \pmod 2 \, \omega$, and if $a \pmod 2 = b \pmod 2 = 1$ then $a + b\omega \pmod 2 = -1 - \omega = -\omega^2$.

Lemma 1. *For a rational prime $q \in \mathbb{Z}[\omega]$, inverse of a nonzero element $z = a + b\omega$ modulo q is given by $|z|^{-2}((a-b) - b\omega)$, where $|z|^{-1}$ is computed modulo integer q in $\mathbb{Z}_q$.*

Proof. Consider $(a + b\omega) * ((a - b) - b\omega) = a^2 - ab + b^2 = |z|^2$, and $|z|^{-1} = |z|^{q-2} \pmod q$ (Fermat's theorem) exists since $\mathbb{Z}_q$ is a field as q is prime integer.

2.4 Group rings

Definition 6 (Group rings). *The group ring of a group $G = \{g_i : i = 1, 2, \ldots, n\}$ over a ring R is the set of formal sums*

$$RG = \left\{ a = \sum_{i=1}^{n} \alpha_i g_i : \alpha_i \in R \text{ for } i = 1, 2, \ldots, n \right\} \tag{7}$$

that forms a ring under the following operations. Suppose $a = \sum_{i=1}^{n} \alpha_i g_i$ and $b = \sum_{i=1}^{n} \beta_i g_i$ in RG.
1. *The sum of a and b is given by $a + b = \sum_{i=1}^{n} (\alpha_i + \beta_i) g_i$.*
2. *The product of a and b is given by $a * b = \sum_{i=1}^{n} \left(\sum_{g_h g_k = g_i} \alpha_h \beta_k \right) g_i$.*

Definition 7 (Coefficient vector). *Every element $a = \sum_{i=1}^{n} \alpha_i g_i$ can be mapped uniquely to its coefficient vector $(\alpha_1, \alpha_2, \ldots, \alpha_n) \in R^n$. We freely use the same notation $'a'$ to denote the elements of the group ring and their corresponding coefficient vectors depending on the context.*

Definition 8 (RG-matrix). *[24] For every element $a = (\alpha_{g_1}, \alpha_{g_2}, \ldots, \alpha_{g_n}) \in RG$, we construct the RG-matrix of a as follows:*

$$M_{RG}(a) = \begin{pmatrix} \alpha_{g_1^{-1} g_1} & \alpha_{g_1^{-1} g_2} & \cdots\cdots & \alpha_{g_1^{-1} g_n} \\ \alpha_{g_2^{-1} g_1} & \alpha_{g_2^{-1} g_2} & \cdots\cdots & \alpha_{g_2^{-1} g_n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{g_n^{-1} g_1} & \alpha_{g_n^{-1} g_2} & \cdots\cdots & \alpha_{g_n^{-1} g_n} \end{pmatrix}. \tag{8}$$

The set $M_{RG} = \{M_{RG}(a) : a \in RG\}$ is the subring of $M_n(R)$. We say a matrix $A \in M_n(R)$ is an RG-matrix if there is an $a \in RG$ such that $A = M_{RG}(a)$.

Theorem 4. *[24, Thereom 1] The mapping $\tau : RG \to M_{RG} \subset M_n(R)$ defined as $\tau(a) = M_{RG}(a)$ is a bijective ring homomorphism.*

Theorem 5. *[24, Thereom 2] An element $a \in RG$ is a unit if and only if $M_{RG}(a)$ is invertible in $M_n(R)$. In that case, inverse of $M_{RG}(a)$ is also an RG-matrix.*

2.5 Group ring $R(C_N \rtimes C_3)$

In this section, we derive some results on the group ring $R(C_N \rtimes_k C_3)$. Particularly, we give the matrix representation of elements in $R(C_N \rtimes_k C_3)$ and then derive an inversion algorithm to check the invertibility and find the inverses of elements in this group ring.

The group ring $R(C_N \rtimes_k C_3)$ can be defined as

$$R(C_N \rtimes_k C_3) = \{\alpha(x) + y\beta(x) + y^2\gamma(x) : \alpha(x), \beta(x), \gamma(x) \in RC_N\}, \tag{9}$$

where $yxy^{-1} = x^k$, $k^3 \equiv 1 \bmod N$. Consider

$$\begin{aligned} (yxy^{-1})^{k^2} &= (x^k)^{k^2} \\ yx^{k^2}y^{-1} &= x \qquad \text{since } k^3 \equiv 1 \bmod N \text{ and } x^N = 1 \\ yx^t &= xy \qquad \text{where } t \equiv k^2 \bmod N. \end{aligned}$$

Therefore,

$$C_N \rtimes_t C_3 = \left\langle x, y \mid x^N = y^3 = 1, xy = yx^t \right\rangle \tag{10}$$

where $3|(N-1), t^3 \equiv 1 \bmod N$, and $t \not\equiv 1 \bmod N$. As a result, $\alpha(x)y = y\alpha(x^t)$ for every $\alpha(x) \in RC_N$. Consequently, the product of two elements $z = u(x) + yv(x) + y^2w(x), a = \alpha(x) + y\beta(x) + y^2\gamma(x) \in R(C_N \rtimes_k C_3)$ is given by

$$\begin{aligned} z * a = u(x)\alpha(x) + w(x^t)\beta(x) + v(x^{t^2})\gamma(x) + y\left(v(x)\alpha(x) + u(x^t)\beta(x) + w(x^{t^2})\gamma(x)\right) \\ + y^2\left(w(x)\alpha(x) + v(x^t)\beta(x) + u(x^{t^2})\gamma(x)\right). \end{aligned} \tag{11}$$

Lemma 2 (Matrix representation). *Let $G = C_N \rtimes_k C_3$ then RG-matrix of an element $z \in R(C_N \rtimes_k C_3)$ is of the form*

$$M_{RG}(z) = \begin{pmatrix} M_0 & M_1 & M_2 \\ M_2 & M_0 & M_1 \\ M_1 & M_2 & M_0 \end{pmatrix} \in R^{3N \times 3N}, \tag{12}$$

i.e., $M_{RG}(z)$ is a block circulant matrix of order $3N$ where each submatrix M_i is an order N matrix.

Proof. We divide the matrix $M_{RG}(z)$ into blocks as

$$M_{RG}(z) = \begin{pmatrix} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \\ A_{20} & A_{21} & A_{22} \end{pmatrix},$$

where A_{rs} is an $N \times N$ matrix over R, for $r, s \in \{0, 1, 2\}$. From the definition 8, for every $0 \le i, j \le N-1$, we have

$$(A_{rs})_{i,j} = \text{ coefficient of } (y^r x^i)^{-1}(y^s x^j) \text{ in } z.$$

Use $xy = yx^t$, where $t \equiv k^2 (\text{mod } N)$, to get $x^i y^j = y^j x^{it^j}$. Further, using $x^N = y^3 = 1$ and $t^3 \equiv 1 \text{ mod } N$, we get

$$(y^r x^i)^{-1}(y^s x^j) \;=\; x^{N-i} y^{(s-r) \text{ mod } 3} x^j \;=\; y^{(s-r) \text{ mod } 3} x^{(j - it^{(s-r) \text{ mod } 3}) \text{ mod } N}.$$

Therefore, $A_{00} = A_{11} = A_{22}$, $A_{01} = A_{12} = A_{20}$, and $A_{02} = A_{10} = A_{21}$. □

Theorem 6 (Units). *Let $z = u(x) + yv(x) + y^2 w(x) \in R(C_N \rtimes_k C_3)$, and $t \equiv k^2 (\text{mod } N)$. Then, z is a unit in $R(C_N \rtimes_k C_3)$ if and only if the element*

$$\begin{aligned} det(u, v, w) = {} & u(x)u(x^t)u(x^{t^2}) + v(x)v(x^t)v(x^{t^2}) + w(x)w(x^t)w(x^{t^2}) \\ & - u(x)v(x^t)w(x^{t^2}) - v(x)w(x^t)u(x^{t^2}) - w(x)u(x^t)v(x^{t^2}) \end{aligned} \tag{13}$$

is a unit in RC_N. In this case, the inverse of z is given by

$$det(u, v, w)^{-1} * \begin{pmatrix} u(x)u(x^t) - v(x^t)w(x^{t^2}) + y(w(x)w(x^{t^2}) - v(x)u(x^{t^2})) \\ +y^2(v(x)v(x^t) - w(x)u(x^t)) \end{pmatrix}. \tag{14}$$

Proof. Element z is a unit if and only if there exists a unique $a = \alpha(x) + y\beta(x) + y^2\gamma(x) \in R(C_N \rtimes_k C_3)$ such that $z * a = a * z = 1$. From (11), we have

$$\begin{aligned} z * a = {} & u(x)\alpha(x) + w(x^t)\beta(x) + v(x^{t^2})\gamma(x) + y\left(v(x)\alpha(x) + u(x^t)\beta(x) + w(x^{t^2})\gamma(x)\right) \\ & + y^2\left(w(x)a(x) + v(x^t)b(x) + u(x^{t^2})\gamma(x)\right) = 1 \end{aligned} \tag{15}$$

Rewriting Equation (15) as

$$\begin{pmatrix} u(x) & w(x^t) & v(x^{t^2}) \\ v(x) & u(x^t) & w(x^{t^2}) \\ w(x) & v(x^t) & u(x^{t^2}) \end{pmatrix} \begin{pmatrix} \alpha(x) \\ \beta(x) \\ \gamma(x) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \tag{16}$$

By uniqueness of inverse, such an a exists if and only if the matrix in Eq. (16) is invertible over RC_N. Consequently, the determinant of this matrix, given precisely by $det(u, v, w)$, is a unit in RC_N. Furthermore, a is obtained as defined in Eq. (14). □

3 GR-NTRU over the group ring $\mathbb{Z}[\omega](C_N \rtimes C_3)$

The Group ring NTRU or GR-NTRU [42] provides a general framework to design NTRU-like cryptosystems by employing different group rings. The standard NTRU operates over the truncated ring of polynomials $\mathbb{Z}[x]/\langle x^N - 1\rangle$. If we let $C_N = \langle x : x^N = 1\rangle$ to be the cyclic group of order N, then $\mathbb{Z}[x]/\langle x^N - 1\rangle$ can be viewed as a group ring of C_N over $\mathbb{Z}$, i.e., $\mathbb{Z}[x]/\langle x^N - 1\rangle \approx \mathbb{Z}C_N$.

Definition 9 (GR-NTRU). *The GR-NTRU generalizes NTRU by replacing the cyclic group ring $\mathbb{Z}C_N$ in NTRU with any group ring $\mathbb{Z}G$ of a finite group G and keeping all other procedures the same with a little modification depending on the requirements.*

3.1 $\mathbb{Z}[\omega](C_N \rtimes C_3)$-NTRU

Let N be a prime number, and $p, q \in \mathbb{Z}[\omega]$ be two primes chosen using Theorem 3 such that $gcd(p, q) = 1$ and $|p| \ll |q|$. We fix $p = 2$ for this work. Our scheme operates over the following rings:

$$R^\omega = \mathbb{Z}[\omega](C_N \rtimes C_3) \text{ and } R^\omega_\alpha = \frac{\mathbb{Z}[\omega]}{\langle\alpha\rangle}(C_N \rtimes C_3), \tag{17}$$

where $\alpha \in \{p, q\}$ and R^ω_α is the set of elements in R^ω whose coefficients are reduced modulo α. Let $r = 2/3$, t be the nearest integer to $r(3N) = 2N$ and s be multiple of 3 nearest to $2N$. The set $L_f \subset R^\omega$ consists of elements with exactly t nonzero coefficients from U_6, and other coefficients are 0. Sets $L_g, L_\phi \subset R^\omega$ consists of elements with $s/3$ triples of coefficients each from sets either U_3 or $-U_3$ in a random order, and other coefficients are 0. The message space is $L_m = R^\omega_p$. In other words, a message is an element of the group ring R^ω whose coefficients belong to the set $D_2 = U_3 \cup \{0\}$. The basic framework of the scheme is similar to NTRU [18] and is sketched as follows:

Key Generation	**Encryption**	**Decryption**
1. Sample $F \xleftarrow{\$} L_f$ until $f = 1 + pF$ is invertible in R^ω_q. 2. $f_q \leftarrow$ inverse of f in R^ω_q. 3. Sample $g \xleftarrow{\$} L_g$. 4. **Public key** $h = f_q * g$ (mod q). 5. **Private key** F.	1. Sample $\phi \xleftarrow{\$} L_\phi$. 2. For messgae $m \in L_m$, compute $e = ph * \phi + m \,(\text{mod} q)$. 3. **return** e .	1. Compute $a = f * e$(mod q). 2. **return** $m = a \,(\text{mod} p)$.

Correctness of decryption. We have $a = p(g * \phi + F * m) + m \,(\text{mod} q)$. If the absolute value of the largest coefficient of $p(g * \phi + F * m) + m$ is less than $|q|/2$, then $a = p(g * \phi + F * m) + m$ without modulo q. Since g, ϕ, and F have maximum $3rN = 2N$ nonzero coefficients and every coefficient has norm 1, also the coefficients of m belong to $U_3 \cup \{0\}$ thus have norm 1. Therefore, the absolute value of the largest possible coefficient of $p(g * \phi + F * m) + m$ is bounded by $4N|p|+1$. So, if we choose q such that $|q| > 8N|p|+2$, then we can eliminate decryption failure entirely. In particular, for $p = 2$, choose q such that $|q| > 16N + 2$.

Inversion. For generating the keys, we need an efficient way to find the inverses of elements in the group ring R^ω_q, where $q \in \mathbb{Z}[\omega]$ is a prime. There exist algorithms [8,37,41] to check the invertibility and find inverses of elements in the ring $\mathbb{Z}_q C_N$ where q is a prime or prime power. These algorithms can easily be modified to work for the ring $\mathbb{Z}[\omega]/\langle q\rangle C_N$. We use the constant-time modular inversion by Bernstein and Yang [8] in our implementation to compute inverses in $\mathbb{Z}[\omega]/\langle q\rangle C_N$ with some modifications as it requires to find inverses in $\mathbb{Z}[\omega]/\langle q\rangle$. That can be done in constant time using the Square-and-Multiply algorithm [38, Page 200] in Lemma 1. Finally, combining the inversion in $\mathbb{Z}[\omega]/\langle q\rangle C_N$ with Theorem 6, one can find invertible elements in the ring R^ω_q as shown in Algorithm 2. The complexity of the inversion algorithm for our scheme and its efficiency over NTRU is discussed in Section 5.

Algorithm 2: Inversion in R_q^ω

Input: $z = u(x) + yv(x) + y^2w(x) \in R_q^\omega$
Output: $z^{-1} = \alpha(x) + y\beta(x) + y^2\gamma(x) \in R_q^\omega$ as inverse of f, or a failure

$d(x) \leftarrow det(u, v, w)$ `/* as in Eq.(13) */`
$inv(x), found \leftarrow$ `find-inverse-of-d(x)-in-`$\frac{\mathbb{Z}[\omega]}{<q>}C_N$
if *not found* **then return** *failure*
$\alpha(x) \leftarrow inv(x) * (u(x^t)u(x^{t^2}) - v(x^t)w(x^{t^2}))$ `/* product in` $\frac{\mathbb{Z}[\omega]}{<q>}C_N$ `*/`
$\beta(x) \leftarrow inv(x) * (w(x)w(x^{t^2}) - v(x)u(x^{t^2}))$ `/* product in` $\frac{\mathbb{Z}[\omega]}{<q>}C_N$ `*/`
$\gamma(x) \leftarrow inv(x) * (v(x)v(x^t) - w(x)u(x^t))$ `/* product in` $\frac{\mathbb{Z}[\omega]}{<q>}C_N$ `*/`
return $z^{-1} = \alpha(x) + y\beta(x) + y^2\gamma(x)$

Probability of decryption failure. Allowing negligible decryption failure in accordance with NIST guidelines can help reduce the key sizes. To model the probability of decryption failure, we follow a similar approach as [26] and make the following assumptions regarding the distribution of coefficients of F, g, ϕ, m.

Assumption 1 *We assume that $r(3N) = 2N$ is evenly divisible by 6 so that the number of nonzero coefficients in g and ϕ is $2N$. Further, assume that all the $2N$ nonzero coefficients of F, g, and ϕ are equi-probable and uniformly distributed over U_6. Similarly, assume all the coefficients of m are uniformly distributed over $U_3 \cup \{0\}$.*

Let $a' = p(g * \phi + F * m) + m$, then the ith coefficient of a' is given by

$$a'_i = p\left(\sum_{j+k\equiv i} g_j\phi_k + \sum_{j+k\equiv i} F_j m_k\right) + m_i$$

for each $0 \leq i \leq 3N$. For a fixed pair (j, k), the terms $g_j\phi_k$ and F_jm_k take the values from the set $U_6 = \{\pm 1, -\frac{1}{2} \pm \frac{\sqrt{3}}{2}\iota, \frac{1}{2} \pm \frac{\sqrt{3}}{2}\iota\}$ each with probabilities $r^2/6 = 2/27$ and $r/8 = 1/12$, respectively. Therefore, the expected mean values of the real and imaginary parts of $g_j\phi_k$ and F_jm_k are zero, i.e., $E(Re(g_j\phi_k)) = E(Im(g_j\phi_k)) = 0$ and $E(Re(F_jm_k)) = E(Im(F_jm_k)) = 0$. Further, their variances are given by

$$Var(Re(g_j\phi_k)) = \frac{r^2}{2} = \frac{2}{9},\ Var(Re(F_jm_k)) = \frac{3r}{8} = \frac{1}{4}.$$

Similarly, $Var(Im(g_j\phi_k)) = 2/9$ and $Var(Im(F_jm_k)) = 1/4$. By the central limit theorem for large N, the real and imaginary parts of a'_i can be modeled as a bivariate normal distribution $(\mathcal{R}, \mathcal{I})$. Then, the means of $\mathcal{R}$ and $\mathcal{I}$ are $\mu_{\mathcal{R}} = \mu_{\mathcal{I}} = 0$ and their variances are

$$\sigma^2 = \sigma_{\mathcal{R}}^2 = \sigma_{\mathcal{I}}^2 = 3Np^2\Big(\frac{r^2}{2} + \frac{3r}{8}\Big) + \frac{3}{8} = \frac{17N}{3} + \frac{3}{8},$$

since, $p = 2$, $E(Re(m_i)) = E(Im(m_i)) = 0$, and $Var(Re(m_i)) = Var(Im(m_i)) = 3/8$, for each i. The probability distribution function for the random variable

$(\mathcal{R}, \mathcal{I})$ at each $x + \iota y$ is $P(x,y) = \frac{1}{2\pi\sigma_{\mathcal{R}}\sigma_{\mathcal{I}}} \exp\left(-\frac{x^2+y^2}{2\sigma_{\mathcal{R}}\sigma_{\mathcal{I}}}\right)$. For successful decryption, we need all the coefficients of a' to be reduced modulo q. Therefore, the probability of successful decryption is given by

$$P_{success}(N,q) = \Big(\iint_{V_q} P(x,y)dxdy\Big)^{3N}. \tag{18}$$

We underestimate the probability of successful decryption to get a closed form of the expression (18):

$$\tilde{P}_{success}(N,q) = \Big(\iint_{C} P(x,y)dxdy\Big)^{3N} = \Big(1 - \exp\Big(-\frac{|q|^2}{8\sigma^2}\Big)\Big)^{3N}, \tag{19}$$

where C is a closed disk of radius $\frac{|q|}{2}$ inscribed inside the voronoi cell V_q. We experimentally confirmed the validity of our model in Figure 2. [1]

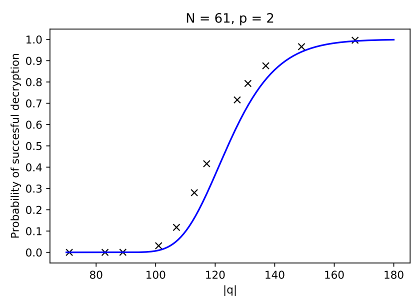


Fig. 2: The probability of successful decryption as a function of $|q|$ for $N = 61, p = 2$. The curve represents $\tilde{P}_{success}(N,q)$, and the crosses represent the ratio of the successful decryption out of $10,000$ randomly generated messages for each prime q.

4 Security analysis

4.1 Combinatorial search attack

Given the public key h and other public parameters, the adversary can try brute force search for some element $f' \in L_f$ such that $f' * h \in L_g$. Therefore, the size of the search spaces is

$$\frac{|L_f|}{3N} = \frac{1}{3N}\binom{3N}{2N}6^{2N}. \tag{20}$$

We have divided by $3N$ to account for all the $3N$ rotations associated with f'. Further, the meet-in-the-middle attack on private key f proposed by Odlyzkoa presented in [20] decreases the size of search space to $\sqrt{|L_f|/3N}$. Table 2 gives the cost (log base 2) of combinatorial or MITM attacks, denoted by **Comb**, against the parameters recommended in Section 5.

[1] The curve in Figure 2 lies slightly below the experimental observations since $\tilde{P}_{success}(N,q)$ (19) gives the underestimated value of the probability of successful decryption while the actual value of our model is given by $P_{success}(N,q)$ (18).

4.2 Lattice attacks

Lattice reduction attacks are the most prominent against NTRU-like schemes. With the knowledge of the public information, the adversary constructs a lattice containing the private key as a short vector that can be recovered by solving SVP or its approximation. First, we discuss the state-of-art cost of lattice reduction algorithms, particularly BKZ, that depends on an important parameter called blocksize β that dominates the runtime. The greater the value of β, the more the runtime and the better the quality of the reduced basis. We call BKZ with blocksize β to be BKZ-β. BKZ has many advancements like [3, 12]. In the literature, many estimators estimate the value of β in higher dimensions. NTRU fatigue estimator [14] is the most accurate one, which is itself based on *2016-estimator* [2]. According to 2016-estimator, for a basis matrix $B = [b_1, b_2, \ldots, b_n]$ of the lattice L_B, BKZ-β detects a unique short vector u if

$$||\pi_{n-\beta}(u)|| < ||b^*_{n-\beta}||, \tag{21}$$

where b^*_i for $i = 1, 2, \ldots, n$ denote the Gram-Schmidt orgthogolization vectors of rows of B, and π_i is the orthogonal projection over $(b_1, b_2, \ldots, b_{i-1})^\perp$. The projected norm is expected to be $\sqrt{\beta/n}||u||$. 2016-estimator adopts the GSA(*Geometric Series Assumption*) [36] that says, for a BKZ reduced lattice with blocksize β, the Gram-Schmidt orthogonalized vectors follow $||b^*_i|| = \delta_\beta^{n-2i-1} det(B)^{\frac{1}{n}}$, where δ_β is called the root Hermite factor of BKZ-β. For $\beta \geq 50$, Chen [11] estimated that

$$\delta_\beta \approx \left(\frac{\beta}{2\pi e}(\pi\beta)^{\frac{1}{\beta}}\right)^{\frac{1}{2(\beta-1)}}. \tag{22}$$

Ducas et al. [14] introduced alternative heuristics, called Z-GSA, for the lengths of Gram-Schmidt vectors of a BKZ-β reduced basis of a q-ary lattice as follows:

Definition 10 (Z-GSA). *[14, Heuristic 2.8] Let B be a basis of a $2n$-dimensional q-ary lattice L_B with n q-vectors. After BKZ-β reduction, the lengths of Gram-Schmidt vectors have the following shape: $m = \frac{1}{2} + \frac{ln(q)}{ln(\delta_\beta)}$ and*

$$||b^*_i|| = \begin{cases} q, & \text{if } i \leq n-m \\ \sqrt{q} \cdot \delta_\beta^{2n-1-2i}, & \text{if } n-m < i < n+m \\ 1, & \text{if } i \geq n+m \end{cases} \tag{23}$$

Since we deal with the lattices of the same nature as described in Z-GSA. Therefore, we employ Z-GSA in 2016-estimation instead of GSA. However, both the models coincide for the successful blocksize when B is a basis of a $2n$-dimensional q-ary lattice with $det(B) = q^n$, as for the lattices in our case.

BKZ uses two approaches to solve SVP: Sieving and Enumeration. Empirical results [29] show that sieving outperforms enumeration starting from a dimension greater than or equal to 65. Therefore, we use the BKZ with Sieving model, denoted as **BKZ(S)**, to compute the cost of lattice attacks. The cost of **BKZ(S)** is modeled as $2^{0.292\beta+o(\beta)}$ (classically) [6] and $2^{0.265\beta+o(\beta)}$ (quantumly) [31].

Primal attack. Gentry [17] introduced a dimension reduction attack on an NTRU variant by factoring the ring $\mathbb{Z}[x]/\langle x^n - 1\rangle$, where n is composite, using the Chinese remainder theorem (CRT). This technique has a possible extension to different algebraic structures, as shown in [30]. Therefore, for any new NTRU-like proposal, it is essential to discuss the possibility of Gentry's attack for a fair security estimate. The underlying algebra in our construction can also be subjected to one layer of Gentry's dimension reduction attack. We discuss the possible homomorphisms that can help the adversary reduce the dimension of the lattice attacks and show that recovering the private key is equivalent to solving SVP in $8N$ dimensional lattices rather than $12N$. However, it is important to point out that the lattices to be attacked in our cryptosystem are difficult to reduce by lattice reduction algorithms in practice.

Notations: We represent every element $a = (\alpha_1 + \omega\beta_1, \dots, \alpha_N + \omega\beta_N) \in \mathbb{Z}[\omega]C_N$ by its integral coefficient vector as $\mathbf{a} = (\alpha_1, \beta_1, \dots, \alpha_N, \beta_N) \in \mathbb{Z}^{2N}$. Similarly, we represent every element $f = (f_0, f_1, f_2) \in \mathbb{Z}[\omega](C_N \rtimes C_3)$ where $f_i \in \mathbb{Z}[\omega]C_N$ by its integral coefficient vector $\mathbf{f} = (\mathbf{f}_0, \mathbf{f}_1, \mathbf{f}_2) \in \mathbb{Z}^{6N}$. For a matrix $A \in M_n(\mathbb{Z}[\omega])$, we define a $2n \times 2n$ integral matrix $\mathbf{A}$ by replacing every entry A_{ij} with a 2×2 integral matrix $\left\langle A_{ij} \right\rangle$ as in (6).

The public key equation can be expressed as

$$\mathbf{f} * \mathbf{H} = \mathbf{g} \,(\mathrm{mod}\, q), \tag{24}$$

where $\mathbf{H} \in \mathbb{Z}^{6N \times 6N}$ is the corresponding integer matrix of R^ω-matrix of the public key h given by $M_{R^\omega}(h) \in M_{3N}(\mathbb{Z}[\omega])$ (Theorem 2). Similar to standard NTRU, the private key $(\mathbf{f}, \mathbf{g})$ can be recovered in a naive way by solving SVP in a 12-dimensional lattice $L_{\mathbf{H}}$ generated by the matrix

$$\mathbf{M_H} = \begin{pmatrix} \mathbf{I}_{6N} & \mathbf{H} \\ \mathbf{0}_{6N} & q\mathbf{I}_{6N} \end{pmatrix}. \tag{25}$$

As discussed in Theorem 2, the matrix $M_{R^\omega}(h)$ and consequently the matrix $\mathbf{H}$ has a special structure

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_1 & \mathbf{H}_2 \\ \mathbf{H}_2 & \mathbf{H}_0 & \mathbf{H}_1 \\ \mathbf{H}_1 & \mathbf{H}_2 & \mathbf{H}_0 \end{pmatrix} \in M_{6N}(\mathbb{Z}) \tag{26}$$

where each $\mathbf{H}_i \in \mathbb{Z}^{2N \times 2N}$. We discuss the scenarios of how an adversary can take advantage of this structure in the context of dimension-reduction attacks. Generally, the goal is to *homomorphically* reduce the size of the public matrix and recover information about the private key that can be lifted back to the original key. Since the value of N is selected to be prime, it rules out the possibility of reducing the size of the matrices $\mathbf{H}_i$ (see [17] for details). The other ring homomorphisms that preserve the information about the private key and prevent the norms of the target vector from growing too large are of the form

$$\mathbf{H} \to \alpha\mathbf{H}_0 + \beta\mathbf{H}_1 + \gamma\mathbf{H}_2 \tag{27}$$

where α, β, γ are small constants. Consequently, it reduces the public key equation to

$$(\alpha\mathbf{f}_0 + \beta\mathbf{f}_1 + \gamma\mathbf{f}_2) * (\alpha\mathbf{H}_0 + \beta\mathbf{H}_1 + \gamma\mathbf{H}_2) = \alpha\mathbf{g}_0 + \beta\mathbf{g}_1 + \gamma\mathbf{g}_2 \pmod q. \tag{28}$$

It can easily be checked that map 27 is a ring homomorphism, i.e., preserve the matrix addition and multiplication, if and only if $(\alpha, \beta, \gamma) \in \{(0,0,0), (1,1,1), (1,\omega,\omega^2), (1,\omega^2,\omega)\}$. The case $(\alpha, \beta, \gamma) = (0,0,0)$ is of no use, therefore, we consider the others only. This way, one is able to reduce the size of the public matrix but end up in matrices with complex entries, apart from when $(\alpha, \beta, \gamma) = (1,1,1)$. In practice, for applying lattice reduction algorithms, such matrices are mapped to the real matrices, which leads to an increase in the dimension. In our case, the matrices

$$\mathbf{H}_{01} = \mathbf{H}_0 + \omega\mathbf{H}_1 + \omega^2\mathbf{H}_2 = (\mathbf{H}_0 - \mathbf{H}_2) + \omega(\mathbf{H}_1 - \mathbf{H}_2),$$
$$\mathbf{H}_{02} = \mathbf{H}_0 + \omega^2\mathbf{H}_1 + \omega\mathbf{H}_2 = (\mathbf{H}_0 - \mathbf{H}_1) + \omega(\mathbf{H}_2 - \mathbf{H}_1)$$

belonging to $M_{2N}(\mathbb{Z}[\omega])$ can be mapped to $4N \times 4N$ integer matrices $\mathcal{H}_{01}$ and $\mathcal{H}_{02}$, respectively, as done before. Suppose $L_{\mathcal{H}_{01}}$ and $L_{\mathcal{H}_{02}}$ are the $8N$-dimensional lattices generated by the matrices

$$\mathbf{M}_{\mathcal{H}_{01}} = \begin{pmatrix} \mathbf{I}_{4N} & \mathcal{H}_{01} \\ \mathbf{0} & q\mathbf{I}_{4N} \end{pmatrix} \quad \text{and} \quad \mathbf{M}_{\mathcal{H}_{02}} = \begin{pmatrix} \mathbf{I}_{4N} & \mathcal{H}_{02} \\ \mathbf{0} & q\mathbf{I}_{4N} \end{pmatrix}. \tag{29}$$

Let $\mathbf{f}_{01}, \mathbf{f}_{02} \in \mathbb{Z}^{4N}$ be the integer vectors corresponding to the element $(\mathbf{f}_0 - \mathbf{f}_2) + \omega(\mathbf{f}_1 - \mathbf{f}_2),\ (\mathbf{f}_0 - \mathbf{f}_1) + \omega(\mathbf{f}_2 - \mathbf{f}_1) \in \mathbb{Z}[\omega]^{2N}$, repspectively. Similarly are defined the vectors $\mathbf{g}_{01},\ \mathbf{g}_{02} \in \mathbb{Z}^{4N}$. Then, the vectors $(\mathbf{f}_{01}, \mathbf{g}_{01}),\ (\mathbf{f}_{02}, \mathbf{g}_{02})$ belong to the lattices $L_{\mathcal{H}_{01}}$ and $L_{\mathcal{H}_{02}}$, respectively. According to Assumption 1, for the secret vector $(\mathbf{f}, \mathbf{g}) = (\mathbf{1} + p\mathbf{F}, \mathbf{g})$, we have

$$||\mathbf{f}_i|| \approx |p| \cdot ||\mathbf{F}_i|| \approx 2\sqrt{\frac{8N}{9}},\ ||\mathbf{g}_i|| \approx \sqrt{\frac{8N}{9}}, \tag{30}$$

and the length of the private vector $(\mathbf{f}, \mathbf{g})$ is approximately $\sqrt{40N/3}$. Therefore,

$$||(\mathbf{f}_{01}, \mathbf{g}_{01})|| \approx ||(\mathbf{f}_{02}, \mathbf{g}_{02})|| \lessapprox \sqrt{\frac{320N}{9}} \approx \sqrt{\frac{8}{3}}||(\mathbf{f}, \mathbf{g})||. \tag{31}$$

While the Gaussian heuristic predicts the length of the shortest vectors in the lattices $L_{\mathcal{H}_{01}}$ and $L_{\mathcal{H}_{02}}$ to be

$$\sigma(L_{\mathcal{H}_{01}}) = \sigma(L_{\mathcal{H}_{02}}) = \sqrt{\frac{4N|q|}{\pi e}} \approx 2.738N. \tag{32}$$

Therefore, the vectors $(\mathbf{f}_{01}, \mathbf{g}_{01})$ and $(\mathbf{f}_{02}, \mathbf{g}_{02})$ are $O(\frac{1}{\sqrt{N}})$ times shorter than the Gaussian expected length. Hence, for large values of N, they are the shortest vectors in the corresponding lattices with a high probability. Thus, the problem of recovering the key is equivalent to solving SVP in $8N$-dimensional lattices that is equal to lattice dimension in the case of NTRU over $\mathbb{Z}C_{N' \approx 4N}$. Conclusively, in our design, the dimension reduction attack reduces the dimension of the lattice by a factor of 1.5, i.e., from $12N$ to $8N$, while DiTRU suffers a dimension loss by a factor of 2. This shows the benefit of working with the semidirect product $C_N \rtimes C_3$.

Hardness of lattice reduction. It is a known fact that the hardness of solving the SVP in a lattice increases with the ratio of the length of the shortest vector to the Gaussian heuristic called *lattice gap* [14]. For NTRU over $\mathbb{Z}C_{N'\approx 4N}$, this ratio[2] is $0.731/\sqrt{N}$, while for our scheme, the lattice gap is $1.54/\sqrt{N}$. Therefore, the lattices associated with our cryptosystem are practically more resistant to lattice attacks compared to the standard NTRU in equal dimensions. To further investigate, corresponding to every parameter set (N', q', p') for NTRU HPS in [35, Table 3], we choose a prime $N \approx N'/4$ and the smallest rational prime $q \in \mathbb{Z}[\omega]$ such that $q > 16N+2$. Then, according to 2016-estimation, we estimate the blocksize β required for recovering the short vectors in lattices $L_{\mathcal{H}_{01}}, L_{\mathcal{H}_{02}}$, and compare with β' required for NTRU in Table 1. It suggests that one can select smaller values of N such that $4N < N'$ for our scheme and still achieve the same security as NTRU over $\mathbb{Z}C_{N'}$.

Table 1: Blocksize estimation for NTRU vs. our scheme for approximately the same dimensions.

NTRU HPS		**Our scheme**	
(N', q', p')	β'	(N, q, p)	β
$(587, 2048, 3)$	456	$(139, 2237, 2)$	506
$(863, 2048, 3)$	701	$(211, 3389, 2)$	777
$(1109, 4096, 3)$	893	$(277, 4451, 2)$	1025

The value of the required blocksize is higher for the new proposal compared to NTRU HPS. It confirms that the lattices associated with our design offer more resistance against lattice reduction techniques, thus resulting in smaller values of N in Table 2.

Hybrid attack. As the name suggests, hybrid attack [21] combines two attacks, the lattice and the combinatorial search attacks. It involves searching for some coefficients of the key in its tail region and reducing a part of the lattice to recover the full secret using the nearest neighborhood algorithm [4]. The parameters of the recent NTRU proposals [9, 25], whose keys are ternary and sparse, are evaluated based on hybrid attacks. However, it is observed that the primal attack outperforms the hybrid attack when the secret key is not ternary, which increases the search cost, as in the case of DiTRU [35]. In our design also, the partial information of the key stored in lower dimensional lattices consists of coefficients from the set $\{0, \pm 1, \pm 2, \pm 3\}$. Expectedly, the overall cost of the hybrid attack exceeds the cost of the primal attack. Therefore, we have selected parameters by considering only the primal attack.

4.3 Overstreched NTRU attack.

An NTRU variant with a very large modulus is referred to as overstretched. The attacks exploiting the presence of specific algebraic structures in overstretched

[2] For NTRU, the length of the key is assumed to be $\sqrt{4N'/3+1}$ and the value of q' that achieves no decryption failure is $q' \geq 16N'/3$. For our scheme, although the norm of the target vector has upper bound $\sqrt{320N/9}$. However, it is empirically observed that the norm of the target vector is approximately $\sqrt{160N/9}$. Therefore, for a conservative estimation of the lattice gap and the blocksize, we consider the latter value of the norm.

NTRU lattices are presented in [1,28]. Later, Ducas and Woerdon [14] narrowed down the estimation on modulus q that separates the overstretched regime from the standard regime. They call this *fatigue point* and showed that for an NTRU lattice of dimension $2n$ with modulus q, the fatigue point is $q \approx 0.004n^{2.484}$. One can verify that the suggested parameter sets in Table 2 for GR-NTRU over $\mathbb{Z}[\omega](C_N \rtimes C_3)$ satisfy $|q| \ll 0.004(4N)^{2.484}$. Therefore, our cryptosystem does not fall under the category of overstretched NTRU and is safe against these kinds of attacks.

5 Parameters and Performance analysis

For our scheme, we are proposing two categories of parameters targeting 128-bit (Level I), 192-bit (Level III), and 256 (Level V) according to NIST definition. Table 2 provides the memory and time requirements for the two selected parameter sets, where the first set provides no decryption failure while the other allows a negligible decryption rate.

Table 2: Parameters for $\mathbb{Z}[\omega](C_N \rtimes C_3)$-NTRU with no decryption failure and negligible decryption failure.

	No decryption failure			**Neglible decryption failure**		
Security level	I	III	V	I	III	V
$(\boldsymbol{N}, \boldsymbol{q}, \boldsymbol{p})$	$(127, 2039, 2)$	$(181, 2903, 2)$	$(241, 3863, 2)$	$(109, 701, 2)$	$(157, 1013, 2)$	$(211, 1361, 2)$
sk (bytes)	153	218	290	131	189	254
pk (bytes)	1143	1629	2350	818	1296	1741
$\boldsymbol{\beta}$	461	664	890	464	663	886
BKZ(S) [classical]	134	193	259	135	193	258
BKZ(S) [quantum]	122	175	235	122	175	234
Comb	505	719	957	433	624	838
Dec failure	–	–	–	2^{-135}	2^{-199}	2^{-269}
CPU cycles $\times 10^3$						
KeyGen	38 163	72 545	131 162	27 498	58 308	103 094
Enc	6 692	11 442	20 452	4 907	9 878	16 313
Dec	12 125	21 308	38 147	8 712	18 109	30 619

Memory requirements. According to [26, Theorem 3], any element $a + b\omega$ reduced modulo q satisfies $a, b \in \left[-2|q|/3, 2|q|/3\right]$. Therefore, the size of the public key $h = f^{-1}g \in \mathbb{Z}[\omega]/\langle q\rangle(C_N \rtimes C_3)$ is $(6N/8) \cdot \left\lceil \log_2 \left(4|q|/3\right) \right\rceil$ bytes. The private key $F \in \mathbb{Z}[\omega](C_N \rtimes C_3)$ is such that its coefficients are of the form $a + b\omega$ where $a, b \in \{0, \pm 1\}$. Since, $-1 \equiv 2 \bmod 3$, and $\sum_{i=0}^{4} 2 \cdot 3^i \leq \sum_{i=0}^{7} 2^i$. Therefore, every five coefficients of F can be stored in 8 bits or 1 byte. Thus, the size of the private key is $\lceil 6N/5 \rceil$ bytes.

Table 3: Memory requirements of the considered NTRU variants.

	NTRU HPS		DiTRU	
Level	**sk**	**pk**	**sk**	**pk**
I	118	808	217	1488
III	173	1187	319	2391
V	221	1664	416	3116

This demonstrates the memory benefits of the proposed scheme as the size of the private (**sk**) and public key (**pk**) (in bytes) of parameters allowing negligible decryption failure for our design are less than DiTRU, while are approximately equal to NTRU HPS.

Performace analysis. In order to analyze the performance of the proposed scheme, we provide a full reference implementation in C. All the provided measurements are evaluated on a single core of 12th Gen Intel(R) Core(TM) i7-1255U with 32 GB RAM and running Linux (Ubuntu 22.04.3 LTS) with TurboBoost and hyper-threading disabled. We compile the code using GCC version 11.4.0-1ubuntu1 22.04 with no optimization flags enabled. Table 2 presents the average CPU cycles required to generate a key, encrypt, and decrypt a message over 10,000 runs. In Table 4, we compare the performance of our work with other prominent NTRU variants in the literature by comparing the CPU cycles needed for key generation and for encrypting/decrypting messages of the same length, **not** only a single message (see Section 5.1). The design rationale of the IND-CCA2 PKE in our work, as well as DiTRU and NTRU HPS, is similar to the one used in the NTRUEncrypt submission [10](see Appendix A).

Table 4: Performance benchmark (*CPU cycles* $\times 10^3$) of this work vs. NTRU and DiTRU for *Key generation*, *Encryption*, and *Decryption* for messages of equal lengths.

	NTRU HPS $(N, q, p = 3)$			**This work** $(N, q, p = 2)$			**Ratio**[a]
	$(587, 2048)$	$(863, 2048)$	$(1109, 4096)$	$(109, 701)$	$(157, 1013)$	$(211, 1361)$	(r_1, r_2, r_3)
Gen:	62 311	146 706	224 363	27 498	58 308	103 094	$(2.27, 2.52, 2.18)$
Enc:	3 132 799	9 105 932	19 790 178	2 772 310	7 569 493	16 294 397	$(1.13, 1.20, 1.21)$
Dec:	5 800 643	17 201 618	37 829 256	4 988 320	13 965 567	30 569 442	$(1.16, 1.23, 1.24)$
	DiTRU $(N, q, p = 3)$						
	$(541, 2048)$	$(797, 4096)$	$(1039, 4096)$	$(109, 701)$	$(157, 1013)$	$(211, 1361)$	(r_1, r_2, r_3)
Gen:	84 756	189 770	308 543	27 498	58 308	103 094	$(3.08, 3.05, 2.99)$
Enc:	9 777 811	29 658 528	66 558 364	5 092 057	14 373 555	30 551 756	$(1.92, 2.06, 2.19)$
Dec:	18 682 243	57 329 287	129 664 570	9 180 125	26 540 407	57 287 299	$(2.04, 2.16, 2.26)$

[a] The ratio is provided as a tuple (r_1, r_2, r_3), where r_1 represents the ratio of CPU cycles needed for key generation, encryption, and decryption by NTRU (DiTRU) to the cycles required by our work in the first level of security. Similarly, r_2 and r_3 represent the ratios measured for the third and the fifth levels of security.

5.1 Discussion

The computational cost of key generation, encryption, and decryption is mainly determined by the 'polynomial' multiplications over the underlying ring. For simplicity, we will discuss the results using the conventional polynomial multiplications. The cost of a polynomial multiplication is $27N^2$ scalar multiplications for this work versus $16N^2$ and $32N^2$ for NTRU and DiTRU, respectively.

Analysis of Key Generation. We discuss the performance of the key generation algorithm when implemented in constant time using the Bernstein-Yang algorithm [8]. For NTRU HPS, the inversion algorithm performs 8 polynomial multiplication in $\mathbb{Z}C_{N'\approx 4N}$ (of cost $\approx 8 \times 16N^2$). Similarly, for DiTRU, the inversion algorithm [35, Algorithm 1] over $\mathbb{Z}D_{N'}$ finds an inverse in $\mathbb{Z}C_{N'}$ plus does 4 extra multiplications over $\mathbb{Z}C_{N'}$, that costs approximately $12 \times 16N^2$. On the other hand, inversion for this work (Algorithm 2) requires 15 multiplications over $\mathbb{Z}[\omega]C_N$ costing $3 \times 15N^2$ scalar multiplications plus an inversion over $\mathbb{Z}[\omega]C_N$. The cost of finding the inverse in $\mathbb{Z}[\omega]C_N$ is upper bounded by $57N^2$ scalar multiplications as detailed in Appendix B. Hence, the cost of constant time implementation of our key generation process is dominated by $102N^2$ scalar multiplications, which is roughly 1.3 and 1.9 faster than NTRU HPS and DiTRU, respectively, when $N' \approx 4N$. In practice, the decryption failure model and a higher lattice gap of this work allow smaller values of $N(< N'/4)$ for equivalent levels of security. As a result, the key generation is roughly two times (three times) faster than the one used in NTRU (DiTRU) in practice. See Table 4.

Analysis of Encryption/Decryption. As in the key generation, the cost of encryption/decryption is dominated by the polynomial multiplications cost. The length of a message encrypted using $\mathbb{Z}C_{N'}$ is $N' \approx 4N$, using $\mathbb{Z}D_{N'}$ is $2N' \approx 8N$, whereas the length of a message encrypted using $\mathbb{Z}[\omega](C_N \rtimes C_3)$ is $6N$ (integer coefficients). Therefore, for a fair comparison of efficiency, we compare the cost of encrypting/decrypting messages of the same length, that is 3 message processings by $\mathbb{Z}C_{N'}$ and 3 message processings by $\mathbb{Z}D_{N'}$ with 2 and 4 message processings, respectively, by $\mathbb{Z}[\omega](C_N \rtimes C_3)$. Therefore, in general, for $N' = 4N$, our cryptosystem is approximately 1.125 times slower than the standard NTRU, while it is approximately 1.7 times faster than DiTRU. However, this is not the case in practice where the parameters selection (considering the smaller value of modulus q and the hardness of the Core-SVP) leads to values of N smaller than $N'/4$. As a result, our cryptosystem is faster than NTRU and DiTRU by approximately a factor of 1.2 and 2, respectively, while encrypting/decrypting messages of the same length. Refer to Table 4.

A Sketched Design Rationale

We follow the same design framework as adopted in NTRUEncrypt [10] to construct a Probabilistic Public Key Encryption (PPKE) scheme. The proposal derives its CPA security from the NTRU assumption, which is transformed into CCA2 secure by employing the NAEP padding mechanism [23]. All the steps in Figure 3 are almost identical to the design rationale used in NTRUEncrypt submission [10], except that the operations are now performed in the noncommutative structure $R^\omega(C_N \rtimes C_3)$ modulo q or p.

KEYGEN($seed$)

1. $g \leftarrow$ `Sampler`$(seed, L_g)$
2. $F \leftarrow$ `Sampler`$(seed, L_f)$
3. $f \leftarrow 1 + pF$
4. `if`(f invertible mod q)
   $f_q \leftarrow$ `inversemodq`(f)
   $h \leftarrow pf_q * g (\bmod q)$
   `return` (h, f)
5. `else` go to step 2

ENCRYPT(h, m)

1. $coins \leftarrow$ `Hash`(h, m)
2. $\phi \leftarrow$ `Sampler`$(coins, L_\phi)$
3. $s \leftarrow \phi * h (\bmod q)$
4. $t \leftarrow$ `Sampler`(`Hash`$(s), L_m)$
5. $m' = m - t (\bmod p)$
6. $c = s + m' (\bmod q)$
7. `return` c

DECRYPT(f, c)

1. $a \leftarrow c * f (\bmod q)$
2. $m' \leftarrow a (\bmod p)$
3. $s \leftarrow c - m' (\bmod q)$
4. $t \leftarrow$ `Sampler`(`Hash`$(s), L_m)$
5. $m \leftarrow m' + t (\bmod p)$
6. `if`(`Encrypt`$(h, m) \neq c$)
   `return` $\perp$
7. `else return` m

Fig. 3: Sketch of the CCA2 secure PPKE for our proposal. The function `Sampler` randomly samples an element unique to the seed from the input space. The spaces L_f, L_g, L_ϕ, and L_m are defined in the Section 3.

B Constant time inversion algorithm for $\mathbb{Z}[\omega]/\langle q \rangle\, C_N$

Algorithm 3: Constant time inversion in $\mathbb{Z}[\omega]/\langle q \rangle\, C_N$

Input: $d(x) \in \mathbb{Z}[\omega]/\langle q \rangle\, C_N$
Output: $delta = 0, inv(x) = d(x)^{-1} \in \mathbb{Z}[\omega]/\langle q \rangle\, C_N$, if $d(x)$ is invertible, else $delta = -1$

$g(x) \leftarrow d(x), f(x) \leftarrow x^N - 1, v(x) \leftarrow 0, r(x) \leftarrow 1$
$delta \leftarrow 1$
for $i = 0$ *to* $2N - 2$ **do**
  $v(x) \leftarrow x * v(x)$
  $swap = (-delta < 0)$ & $(g_0 \neq 0)$
  $delta$^ $=$ $swap$ & ($delta$^ $- delta$)
  $delta = delta + 1$
  `constSwap`$(f(x), g(x), swap)$ `/* swap` $f(x)$ `and` $g(x)$ `if` $swap$ `is 1 */`
  `constSwap`$(v(x), r(x), swap)$ `/* swap` $v(x)$ `and` $r(x)$ `if` $swap$ `is 1 */`
  $g(x) \leftarrow f_0 g(x) - g_0 f(x) (\bmod q)$
  $r(x) \leftarrow f_0 r(x) - g_0 v(x) (\bmod q)$
  $g(x) \leftarrow g(x)/x$
$k \leftarrow$ `inverse-mod q-in-`$\mathbb{Z}[\omega](f_0)$ `/* inverse of` f_0 `in` $\mathbb{Z}[\omega]$ `modulo` q `*/`
$inv(x) \leftarrow k*$`reverse`$(v(x))(\bmod q)$ `/* reverse coefficients of` $v(x)$ `*/`
return $delta, inv(x)$

Algorithm 3 is a direct adaptation of the Bernstein-Yang algorithm [8] with the required modifications to our new ring $\mathbb{Z}[\omega]/\langle q \rangle\, C_N$.

– Multiplication of two Eisenstein integers requires 3 integer multiplications.
– Modulo q in $\mathbb{Z}$ (for a prime q) requires 4 scalar multiplications in constant-time, therefore modulo q in $\mathbb{Z}[\omega]$ (for a prime $q + 0\omega$) requires 8 scalar multiplications (Algorithm 1).
– Inversion of an element in $\mathbb{Z}[\omega]$ modulo q is upper bounded by $17+10\log_2(q-2)$ scalar multiplications as in Lemma 1.

Therefore, lines 10 and 11 contribute to $14(N+1)$ scalar multiplications each. Line 13 contributes to $17+10\log_2(q-2)$ multiplications, and line 14 contributes to $11N$ scalar multiplications.

Acknowledgments. The authors want to express their gratitude to the reviewers whose valuable suggestions have greatly helped improve the editorial quality of the paper. Vikas Kumar would like to thank CSIR for supporting his research through grant no. 09/143(1038)/2020-EMR-I.

References

1. Albrecht, M., Bai, S., Ducas, L.: A subfield lattice attack on overstretched NTRU assumptions. In: Advances in Cryptology – CRYPTO 2016. pp. 153–178. Springer Berlin Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_6
2. 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
3. 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
4. Babai, L.: On Lovász' lattice reduction and the nearest lattice point problem. Combinatorica **6**, 1–13 (1986), https://doi.org/10.1007/BF02579403
5. 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
6. 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
7. 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
8. 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
9. 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
10. Chen, C., Hoffstein, J., Whyte, W., Zhang, Z.: NIST PQ Submission: NTRUEncrypt A lattice based encryption algorithm. NIST (2017), https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions

11. Chen, Y.: Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe. Ph.D. thesis, l'Université Paris Diderot (2013), http://www.theses.fr/2013PA077242
12. 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
13. 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
14. 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
15. Dummit, D.S., Foote, R.M.: Abstract Algebra. John Wiley & Sons, Inc., 3 edn. (06 2003), https://www.wiley.com/en-in/Abstract+Algebra%2C+3rd+Edition-p-9780471433347
16. Fox, N.: Spectra of Semidirect Products of Cyclic Groups. Rose-Hulman Undergraduate Mathematics Journal **11** (2010), https://scholar.rose-hulman.edu/rhumj/vol11/iss2/7
17. 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
18. Hoffstein, J., Pipher, J., Silverman, J.: An Introduction to Mathematical Cryptography. Springer Publishing Company, Incorporated, NY, 1 edn. (2008)
19. 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
20. 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
21. 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
22. Howgrave-Graham, N., Nguyen, P.Q., Pointcheval, D., Proos, J., Silverman, J.H., Singer, A., Whyte, W.: The impact of decryption failures on the security of ntru encryption. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003. pp. 226–246. Springer Berlin Heidelberg, Berlin, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_14
23. Howgrave-Graham, N., Silverman, J.H., Whyte, W.: Choosing parameter sets for NTRUEncrypt with NAEP and SVES-3. In: Menezes, A. (ed.) Topics in Cryptology – CT-RSA 2005. pp. 118–135 (2005). https://doi.org/10.1007/978-3-540-30574-3_10
24. 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
25. Hülsing, A., Rijneveld, J., Schanck, J., Schwabe, P.: High-speed key encapsulation from NTRU. In: International Conference on Cryptographic Hardware and Embedded Systems, CHES 2017. pp. 232–252 (2017). https://doi.org/10.1007/978-3-319-66787-4_12

26. Jarvis, K., Nevins, M.: ETRU: NTRU over the eisenstein integers. Des. Codes Cryptogr. **74**, 219–242 (2015), `https://doi.org/10.1007/s10623-013-9850-3`
27. Kim, J., Lee, C.: A polynomial time algorithm for breaking NTRU encryption with multiple keys. Designs, Codes and Cryptography **91**, 2779–2789 (2023). `https://doi.org/10.1007/s10623-023-01233-5`
28. 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`
29. Kirshanova, E., May, A., Nowakowski, J.: New NTRU Records with Improved Lattice Bases. In: Post-Quantum Cryptography. pp. 167–195 (2023). `https://doi.org/10.1007/978-3-031-40003-2_7`
30. Kumar, V., Raya, A., Gangopadhyay, S., Gangopadhyay, A.K.: Lattice attack on group ring NTRU: The case of the dihedral group (2023), `https://doi.org/10.48550/arXiv.2309.08304`
31. 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`
32. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische annalen **261**(ARTICLE), 515–534 (1982). `https://doi.org/10.1007/BF01457454`
33. Malekian, E., Zakerolhosseini, A.: OTRU: A non-associative and high speed public key cryptosystem. In: 2010 15th CSI International Symposium on Computer Architecture and Digital Systems. pp. 83–90 (2010). `https://doi.org/10.1109/CADS.2010.5623536`
34. 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`
35. 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`
36. Schnorr, C.: A hierarchy of polynomial time lattice basis reduction algorithms. Theoretical Computer Science **53**(2), 201–224 (1987). `https://doi.org/https://doi.org/10.1016/0304-3975(87)90064-8`
37. Silverman, J.H.: Almost inverses and fast NTRU key creation. NTRU Cryptosystems Technical Report #14 (1999), `https://ntru.org/f/tr/tr014v1.pdf`
38. Stinson, D., Paterson, M.: Cryptography: Theory and Practice. CRC Press, Chapman and Hall Book, Taylor & Francis, 4 edn. (2017), `https://doi.org/10.1201/9781315282497`
39. Thakur, K.: A variant of NTRU with split quaternions algebra. Palestine J. of Mathematics **6**(2), 598–610 (2017), `https://pjm.ppu.edu/sites/default/files/papers/PJM_April_2017_28.pdf`
40. Truman, K.R.: Analysis and Extension of Non-Commutative NTRU. PhD dissertation, University of Maryland (2007), `https://drum.lib.umd.edu/handle/1903/7344`
41. Venier, D., Cheung, R.C.: A highly parallel constant-time almost-inverse algorithm. In: 2020 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC). pp. 1–6 (2020). `https://doi.org/10.1109/ICSPCC50002.2020.9259505`

42. 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