# Efficient implementation of post quantum MLWR-based PKE scheme using NTT

**Article** *in* Computers & Electrical Engineering · June 2024

DOI: 10.1016/j.compeleceng.2024.109358

**2 authors:**

Anupama Arjun Pandit
Defence Institute of Advanced Technology
5 PUBLICATIONS 11 CITATIONS
SEE PROFILE

Arun Mishra
Defence Institute of Advanced Technology
38 PUBLICATIONS 125 CITATIONS
SEE PROFILE

# Efficient Implementation of Post Quantum MLWR-based PKE Scheme using NTT

Anupama Arjun Pandit[a], Arun Mishra*[a]

[a]*Department of Computer Science and Engineering, SoCE&MS, Defence Institute of Advanced Technology, Girinagar Pune, 411025, Maharashtra, India*

**Abstract**

Conventional Public-Key Encryption (PKE) techniques are susceptible to possible attacks by quantum computers. Lattice-based Cryptography presents promising solutions with quantum-safe hard problems like learning with rounding, short integer solutions, learning with errors, and Module Learning with Rounding (MLWR). Proposed work presents an efficient MLWR-based PKE scheme that utilizes number-theoretic transforms to significantly reduce polynomial multiplication time up to 40%. Additionally, scheme uses rounding operations to reduce public key size. Proposed scheme offers a balance between security and efficiency. Key generation, encryption, and decryption operations requiring 1,422, 1,040, and 2,647 CPU cycles respectively and execution times for these operations are approximately 68.965, 34.483, and 34.483 $\mu$s. The scheme's security against chosen plaintext attacks is proven under the MLWR assumption. Security assessments using Martin Albrecht's estimator demonstrate that proposed parameter sets offer a high level of security (NIST's security categories) against Block Korkine-Zolotarev (BKZ) attacks. A comprehensive evaluation of the proposed PKE scheme has been conducted, focusing on its performance and security. The analysis includes a comparison with other lattice-based PKEs to highlight scheme's unique features and advantages. Performance of proposed PKE is comparable to existing efficient lattice-based alternatives, making it a compelling candidate for secure communication in the post-quantum era.

*Keywords:* Module-Learning with Rounding, Post-Quantum Cryptography, NTT, Learning with Error, Lattice based Cryptography, Public-Key Encryption

## 1. Introduction

Security of existing public key cryptosystems is based on classical hard problems that are exponentially hard for classical computers to solve. However, the development of quantum computers threatens the security of these systems as they can solve classical hard problems in polynomial time using Shor's algorithm [1]. To address quantum threats, the National Institute of Standards and Technology (NIST) has started an initiative aimed at creating cryptographic primitives for Post-Quantum Cryptography (PQC) [2]. Goal of PQC project is to develop novel cryptographic algorithms that demonstrate resilience against quantum computer attacks and can provide long-term security for sensitive

*Email address:* `arunmishra@diat.ac.in` (Arun Mishra* ) data and communication. As part of the PQC project, NIST has selected four Public Key Encryption (PKE) candidates as third-round finalists. These candidates have been chosen based on their security, efficiency, and practicality for real-world applications. Three of the four finalists - SABER [3], NTRU [4], CRYSTALS-Kyber [5] are lattice-based schemes.

Lattice-based cryptography (LBC) has been explored over the last decade as it has advantages of fast implementation, strong security, quantum resistance, and efficiency [2]. Due to the small key size and ease of implementation, LBC is suitable for the efficient implementation of encryption schemes. Most of these schemes are quantum-safe lattice-based candidates.

LBC is constructed using lattice-based hard problems like Learning with Rounding (LWR) [6], Learning with Errors (LWE) [7], Ring LWR (RLWR) [8], Module LWE (MLWE) [5], and Module LWR (MLWR) [8].

In 2005, Oded Regev came up with the LWE problem [7] to design PKE. The security of LWE against quantum computers has led to its widespread application in many cryptographic schemes. Security of LWE is based on its reduction from the worst-case lattice problem, which is believed to be resistant to quantum computer attacks. The encryption algorithms based on LWE are computationally expensive as they involve fundamental processes like discrete Gaussian sampling and polynomial computations. LWR problem [6] presented by Banerjee A. et.al., is a derandomized version of LWE, where the random error term in LWE is replaced with a deterministic rounding operation. In LWR output value is rounded to its closest element in a sufficient course subset. LWR is used to construct Pseudo Random Number Generators (PRNGs) and has the same hardness as the LWE problem with uniform noise distribution [9]. If the number of samples were bound, the LWR problem would be as hard as the LWE problem [8], and distinguishing any LWR sample from a random sample is as hard as the LWE problem with uniform noise distribution.

To efficiently implement lattice-based schemes, ring setups and ideal lattices [8] are most commonly used. Majority of the NIST finalists are "ring/ideal lattice schemes", that are based on polynomial rings as their core algebraic structure. Polynomial multiplication modulo a specific polynomial is a critical algebraic operation in these schemes. To achieve efficient polynomial multiplication over an integer ring, ring LBC schemes make use of the Number Theoretic Transform (NTT) technique.

Module lattices [10] lie between ideal and unstructured lattices. These are constructed based on modules of the form $R_q^k$ (where k is a rank of a module). Module-based constructions offer comparable efficiency to ideal lattices while providing multiple advantages for practical applications. Existing PKE schemes that are based on MLWR utilize module lattices as their fundamental mathematical framework. These schemes take the benefits of module lattices and integrate them with rounding methods to attain encryption schemes that are both secure and efficient. Notably, MLWR-based schemes proposed by D'Anvers et al. and others [3] utilise algorithms such as School Book [11], ToomCook, and Karatsuba [12] algorithms for polynomial multiplication. In order to improve the MLWR-based PKE schemes, it is essential to integrate an efficient polynomial multiplication approach such as NTT.

***Our Contributions***

The present work proposes an efficient software implementation of a PKE scheme based on MLWR with NTT.

- **Module LWR for PKE**

  The proposed approach has been implemented without using Gaussian noise sampling, as MLWR instances do not necessitate it. Utilization of MLWR achieves a balance between security and efficiency compared to the original LWR and RLWR problems [8]. Moreover, the MLWR hardness assumption demonstrates INDistinguishability under Chosen Plaintext Attack (IND-CPA) security of the proposed MLWR-based PKE scheme. Benefits of software implementation include a shortened design cycle and high flexibility. Module-lattice structure offers flexibility by reusing a single core component for different levels of security.

  Parameter sets are proposed for different security levels based on the best-known attacks against LWR. To mitigate potential vulnerabilities from known attacks on LWR, the proposed scheme selects parameter sets carefully to provide the desired level of security.

- **Efficient implementation**

  In proposed scheme, NTT [13] method is used to increase the speed of matrix multiplication. NTT is implemented with negative wrapped convolution [14] which decreases output of the multiplication by $1 + x^N$ without the need for additional memory. NTT implementation used is a natural iterative implementation with 32-bit unsigned integers. It incorporates Cooley-Tukey butterflies and Gentleman-Sande butterflies for forward and inverse transforms, respectively [13]. This implementation reduces computation time to $O(NlogN)$.

  Proposed PKE approach includes algorithms for Key Generation, Encryption, and Decryption. Generalized hash function Shake-128 (eXtendable Output Function (XOF)) [15] is employed as an expansion function to generate a public matrix. The time required to generate keys and decryption process is significantly less compared to other efficient lattice-based PKEs. However, encryption time is equivalent to the most prominent NIST candidates. Reported parameters are selected as per the revised LWR properties explained by Joel Alwen et.al. [16].

**Organisation of the paper**

Section 2 covers a comprehensive summary of related research in the field of lattice-based PKE schemes. Section 3 presents basic notations and provides background information on lattice-based hard problems. Section 4 provides details of the proposed MLWR-based PKE scheme, including its implementation, security proofs, and correctness details. Section 5 presents a comprehensive analysis of the proposed scheme's performance and security. Section 6 provides a comparison between the proposed PKE and other lattice-based PKE approaches. Finally, Section 7 concludes the paper by summarizing the contributions made by proposed work and outlining potential future directions.

## 2. Related Work

LBC has gained attention as a suitable candidate for efficient and lightweight cryptography. Security of LBC is ingrained in worst-case hard problems, such as Shortest Vector Problem (SVP) [17], Closest Vector Problem [18], GapSVP [19], approximate Shortest Independent Vectors Problem (SIVP) [10], LWE [7], and Small Integer Solution (SIS) problem [10]. These problems provide cryptographically worst-case hardness guarantees.

Early lattice-based PKE schemes were based on SVP. In 1996, Ajtai and Dwork introduced lattice-based PKE cryptosystems that demonstrated security based on worst-case complexity assumptions [20]. However, these schemes required high-dimensional lattices, resulting in large key sizes in gigabits. Breaking these schemes is as hard as solving the hidden hyperplanes problem [18].

In 1998, J. Hoffstein et al. proposed the NTRU asymmetric cryptographic system based on polynomial rings [21]. NTRU encryption offers compact key sizes and resistance against cryptanalytic efforts. NTRU variants have been proven secure under the hardness assumptions of Ring LWE(RLWE)[18].

The LWE-based PKE scheme, first presented by Regev et al. [7] in 2005, was more efficient than previous lattice-based PKE schemes. Regev's scheme has smaller key and ciphertext sizes compared to earlier LWE-based PKE schemes. However, it is only secure against passive attacks and has limited resistance against active attacks or chosen plaintext attacks (CPA) [10]. In 2011, Richard Lindner and Chris Peikert gave PKE [22] with smaller secret keys and ciphertext than earlier LWE-based PKE's by a factor of approximately *log q*.

In 2016, NIST requested submissions for PQC algorithms, including lattice-based algorithms, for standardization [2]. Several lattice-based PKE algorithms were selected in the $2^{nd}$ round, including Frodo, Kyber, and SABER, which are efficient and promising candidates for post-quantum security. A review of the lattice-based PKE algorithms is as follows:

**Frodo**[23] is a Key Encapsulation Mechanism (KEM) that offers smaller keys and ciphertext sizes compared to Regev's scheme. It generates a public square matrix using a small seed, resulting in compact keys. However, Frodo is less efficient than schemes based on algebraically structured versions of LWE, and its bandwidth usage and execution times are higher. Most time-consuming arithmetic operation in Frodo is a multiplication of matrices with entries in $\mathbb{Z}_q$ .

**Kyber's**[5] security is based on the hardness of the MLWE problem. It uses $R_q$ and a Centered Binomial Distribution (CBD) [13] to sample secrets and errors. Kyber offers optimized implementations that focus on Keccak and NTT arrangements, which enhance its performance. NTT-based multiplication provided by Kyber parameters is memory-efficient and faster than alternative methods for polynomial ring multiplication. Kyber has been selected as a finalist by NIST and is considered a strong candidate for standardization.

**Saber** [3] is based on the hardness of a MLWR problem. MLWR reduces bandwidth and requires less randomness compared to MLWE-based designs. Saber uses MLWR, eliminating the need for error sampling. It requires fewer secret polynomials compared to MLWE schemes, resulting in increased efficiency. Saber employs power-of-two moduli and efficient polynomial arithmetic techniques such as Toom-4 and Karatsuba. While power-of-two moduli are incompatible with simple NTTs, it is still feasible to employ NTTs to construct their underlying polynomial arithmetic and achieve higher performance when compared to state-of-the-art implementations.

Proposed PKE approach aims to improve efficiency by using a combination of NTT and MLWR. The use of NTT reduces computation complexity, as demonstrated in Kyber, while MLWR reduces bandwidth requirements and randomness compared to MLWE, as observed in Saber. By leveraging benefits of both NTT's and MLWR's benefits, proposed PKE scheme aims to provide an efficient lattice-based encryption solution.

## 3. Preliminaries

### *3.1. Notations*

In proposed work, certain notations and definitions are used to represent various elements and concepts. Matrices are represented using uppercase bold alphabets, such as $\mathbf{A}$, scalars by lowercase regular alphabets, such as $p$, and vectors are represented using lowercase bold alphabets, such as vector $\mathbf{s}$. Notation $\mathbf{s} \leftarrow U_0$ indicates that the variable $\mathbf{s}$ is sampled according to the distribution $U_0$. When $U_0$ is a finite set, this denotes uniform sampling. Transpose of a matrix $\mathbf{A}$ is denoted as $\mathbf{A}^T$. Lattice constructed using basis set $B$ is represented by symbol $\mathcal{L}(B)$. Rounding of a real number $r$ is denoted by $\lfloor r \rceil = \lfloor (p/q) \cdot r \rceil \bmod p$, where $p$ is a rounding modulus. Ring of integers under modulus $q$ is represented using $\mathbb{Z}_q$. Polynomial ring $\mathbb{Z}_q[X]/(X^N + 1)$ is represented by $R_q$, where $N$ is the dimension of the ring. In the context of rings, the notation $R^{l \times l}$ represents the ring of $l \times l$ matrices over the ring $R$. In module lattices $R^{k \times l}$, $k$ represents the rank of the module. Twiddle factors [13] Fast Fourier Transform (FFT) process are expressed mathematically as: $\omega_N^n = e^{-i2\pi n/N} = cos(2\pi n/N) - i\ sin(2\pi n/N)$, where $n = 0, 1, 2 \cdots, N-1$. Private keys and public keys are denoted by $pk$ and $sk$ respectively. Symbol $\delta$ represents the seed.
IND-CPA is a security notion used to evaluate cryptographic schemes. Table 1 provides details about the security levels defined by NIST for PQC [24].

Table 1: Security Levels

| Security Level | Security Requirements* |
|---|---|
| 1 | *key* search on 128-bit (AES128) block cipher |
| 2 | collisions searching on 256-bit hash function (SHA3-256/SHA256) |
| 3 | *key* search on 192-bit (AES192) block cipher |
| 4 | collisions searching on 384-bit hash function (SHA3-384/SHA384) |
| 5 | *key* search on 256-bit (AES256) block cipher |

* An attack which violates its applicable security standard should necessitate computing resources equivalent to or higher than that required for *key* search on AES and collisions on hash functions.

### *3.2. Hardness Assumption: LWE, LWR and MLWR problems*

- **LWE**: The decision version of the LWE problem [7] states that it is difficult to distinguish between uniform random samples $(\mathbf{a}, u) \Leftarrow (\mathbb{Z}_q^m \times \mathbb{Z}_q)$ and LWE-samples of the form

$$\mathbf{a}_{n \times m}\mathbf{s}_{m \times 1} + \mathbf{e}_{n \times 1} = b_{n \times 1} mod q \in (\mathbb{Z}_q^m \times \mathbb{Z}_q)$$

where, **a** and **s** are chosen uniformly at random, $b =< \mathbf{a}, \mathbf{s} > +\mathbf{e}$ *mod q* is independent of **a**, and it represents a noisy inner product of **a** and random secret $\mathbf{s} \in \mathbb{Z}_q^m$ with addition of small error chosen randomly from uniform distribution [7] $\chi$ is $\mathbf{e} \in \chi$.

- **LWR**: Decision LWR problem [6] states that it is difficult to distinguish LWR distribution samples from uniform distribution samples. In, LWR secret vector $\mathbf{s} \in \mathbb{Z}_q^m$ and LWR samples are generated by rounding the inner product of **a** and random secret **s**. It is shown by

$$(\mathbf{a}, b = \lfloor < \mathbf{a}, \mathbf{s} > \rceil_p) \in \mathbb{Z}_q^m \times \mathbb{Z}_p$$

- **MLWR**: Proposed scheme's security is based on the MLWR problem. Hardness of MLWR is a simple extension of MLWE. An MLWR sample is represented as in eq. 1.

$$(\mathbf{a}, b = \lfloor < \mathbf{a}, \mathbf{s} > \rceil_p) \in R_q^m \times R_p \tag{1}$$

Hardness of MLWR relies on the assumption that LWR is hard, and according to authors [6, 3] LWR has proven to be at least as hard as LWE. Reduction of LWE to LWR depends on the fact that $\lfloor < \mathbf{a}, \mathbf{s} > +\mathbf{e} \rceil_p = \lfloor < \mathbf{a}, \mathbf{s} > \rceil_p$ where error e is derandomized by $\frac{q}{p}$. Also $\lfloor U(\mathbb{Z}_q) \rceil_p = U(\mathbb{Z}_p)$ where $U$ represents uniform distribution. Hence, for any stated samples $< \mathbf{a}_i, b_i >$ which belongs to LWE or uniform type, $b_i$ terms could be rounded off to produce samples belonging to LWR or uniform type. However, it is also possible that $\lfloor < \mathbf{a}, \mathbf{s} > + e \rceil_p \neq \lfloor < \mathbf{a}, \mathbf{s} > \rceil_p$ because of the error $e$.
The hardness of the MLWR problem needs the modulus $q$ to be superpolynomial in relation to the bounded error distribution. This ensures that the MLWR problem is hard only when $q > p$, where $p$ is rounding modulus. The error would be approximately $\frac{q}{p}$.

### 3.3. *Polynomial Arithmetic and NTT*

Proposed system utilizes NTT for polynomial arithmetic and multiplication. The selection of NTT-based polynomial multiplication is based on its efficiency, with a time complexity of $O(NlogN)$ [13], where $N$ represents the degree of a polynomial. NTT algorithm used in the scheme employs negative wrapped convolution [14], which allows for reducing output of the multiplication by $1 + x^N$ without using extra memory.
Modulus $q$ in the proposed implementation is selected so that a $2N^{th} = (2 \times 256)^{th} = 512^{th}$ root of unity $\omega$ modulo $q$ will exist. Specifically, a value of $\omega = 1753$ is commonly used. This selection ensures that the cyclotomic polynomial $X^N + 1$ can be decomposed into linear factors $X - \omega^j$ modulo $q$, where $N = 256$ and $j$ takes values $0, 1, 3, 5, 7, ..., 511$. Chinese Remainder Theorem (CRT) [25] is applied to show that the cyclotomic ring $R_q$ is isomorphic to the product of rings $\mathbb{Z}_q[X]/(X - \omega^j) \approx \mathbb{Z}_q$. This isomorphism allows simple multiplication of elements (pointwise multiplication) in the ring product.

$$pl \to (pl(\omega), pl(\omega^3), pl(\omega^5), \cdots pl(\omega^{511}) : R_q \to \prod_j \mathbb{Z}_q[X]/(X - \omega^j)$$

Isomorphism can be efficiently calculated using the FFT [26]. By exploiting the below relationship

$$X^{256} + 1 = X^{256} - \omega^{256} = (X^{128} + \omega^{128})(X^{128} - \omega^{128})$$

$$\therefore \qquad \mathbb{Z}_q[X]/(X^{256} + 1) \to (\mathbb{Z}_q[X]/(X^{128} + \omega^{128}))(\mathbb{Z}_q[X]/(X^{128} - \omega^{128}))$$

calculation can be done independently on two reduced polynomials with degrees 128, considering that $X^{128} - \omega^{384} = X^{128} + \omega^{128}$. This approach enables fast NTT algorithms, which are essentially defined as FFT in the finite field case. Generally, fast NTT algorithms don't produce vector coefficients of order $pl(\omega), pl(\omega^3), pl(\omega^5), ..., pl(\omega^{511})$.
In the NTT domain representation, polynomial $pl \in R_q$ is denoted as its NTT transform, denoted as $\hat{pl} = NTT(pl) \in \mathbb{Z}_q^{256}$. Coefficients of $\hat{pl}$ correspond to the evaluations of $pl$ at specific values $\omega_j$
$\hat{pl} = NTT(pl) = (pl(\omega_0), pl(-\omega_0), ..., pl(\omega_{126}), pl(-\omega_{126}), pl(\omega_{127}), pl(-\omega_{127}))$
where, $\omega_j = \omega^{brv(128+j)}$ and $brv(v)$ represents bit-reversal of value $v$ with 8 bits. NTT domain representation allows for efficient polynomial multiplication, with a product of polynomials $pl_1$ and $Pl_2$ given as
$pl_1 Pl_2 = NTT^{-1}(NTT(pl_1) \bigodot NTT(pl_2))$ where $\bigodot$ represents element-wise multiplication.

### 3.3.1. Negative Wrapped Convolution (NWC)

NWC [26] technique is used to perform multiplication of two polynomials over the ring $\mathbb{Z}_q[x]/\langle x^N + 1\rangle$. NWC technique is applied as follows:
Let $\bar{x}_i = x_i\gamma_{2N}^i, \bar{y}_i = y_i\gamma_{2N}^i$, and $\bar{z}_i = z_i\gamma_{2N}^i$ where $\gamma_{2N} = \sqrt{\omega_N}$.
To compute $z = x \cdot y$ over $\mathbb{Z}_q[x]/\langle x^N + 1\rangle$, NWC technique [26] is performed as:
$\bar{z} = NTT_N^{-1}(NTT_N(\bar{x}) \bigodot NTT_N(\bar{y}))$
If $q \equiv 1 \ (mod\ 2N)$ is satisfied by prime $q$, then $\omega_N$ and its square root $\gamma_{2N}$ exist, as per NWC technique. This approach, however, has a "scramble" that includes pre-processing before NTT and post-processing after Inverse NTT (NTT$^{-1}$), as discussed below. Scaled vectors $\bar{x}$ and $\bar{y}$ are subjected to classic N-point NTT; scaled vector $\bar{z}$ is produced after the classic N-point NTT$^{-1}$; and final result $z$ may be retrieved by computing $z_i = \gamma_{2N}^{-i}$. NWC technique [26] eliminates explicit reduction and doubling of NTT / NTT$^{-1}$, but it does need coefficients to be scaled with $\gamma_{2N}^i$ before NTT and results to be scaled with $\gamma_{2N}^{-i}$ after NTT$^{-1}$.

NTT algorithm without pre-processing [27] is as follows:
Suppose vectors $v$ and $V$ indicate $v_0, v_1, \cdots, v_{N-1}$ and $(V_0, V_1, \cdots, V_{N-1})$ respectively, where $v_i \in \mathbb{Z}_q, V_i \in \mathbb{Z}_q, i = 1, 2, 3, \cdots, N-1$. Suppose $\omega_N$ be a primitive $N^{th}$ root of unity in $\mathbb{Z}_q$ and $\gamma_{2N} = \sqrt{\omega_N}$.
$Input : v, N, q, \gamma_{2N}^i, i = 0, 1, 2, \cdots N-1$
$Output : V = NTT(v)$

```
i $V \leftarrow scramble(v)$
```

```
ii   $for\ l = 1\ to\ \log_2 N\ do$
iii      $r \leftarrow 2^l$
iv       $for\ m = 0\ to\ r/2 - 1\ do$
v            $\omega = \gamma_{2N}^{(2m+1)N/r}$
vi           $for\ n = 0\ to\ N/r - 1\ do$
vii              $f = V_{nr+m}$
viii             $g = \omega \cdot V_{nr+m+r/2}\ mod\ q$
ix               $V_{nr+m} = (f + g)\ mod\ q$
x                $V_{nr+m+r/2} = (f - g)\ mod\ q$
xi           $end\ for$
xii      $end\ for$
xiii $end\ for$
xiv  $return\ V$
```

NTT$^{-1}$ algorithm without post-processing [26] is as follows:

Suppose vectors $v$ and $V$ indicate $v_0, v_1, \cdots, v_{N-1}$ and $(V_0, V_1, \cdots, V_{N-1})$ respectively, where $v_i \in \mathbb{Z}_q, V_i \in \mathbb{Z}_q, i = 1, 2, 3, \cdots, N-1$. Suppose $\omega_N$ be a primitive $N^{th}$ root of unity in $\mathbb{Z}_q$ and $\gamma_{2N} = \sqrt{\omega_N}$.

$Input : v, N, q, \gamma_{2N}^{-i}, i = 0, 1, 2, \cdots N-1$

$Output : V = NTT^{-1}(v)$

```
i    $for\ l = \log_2 N\ to\ 1\ do$
ii       $r \leftarrow 2^l$
iii      $for\ m = 0\ to\ r/2 - 1\ do$
iv           $\omega = \gamma_{2N}^{-(2m+1)N/r}$
v            $for\ n = 0\ to\ N/r - 1\ do$
vi               $f = V_{nr+m}$
vii              $g = V_{nr+m+r/2}\ mod\ q$
```

```
viii                $V_{nr+m} = \frac{(f+g)}{2}\ mod\ q$
ix                  $V_{nr+m+r/2} = \frac{(f-g)}{2} \cdot \omega\ mod\ q$
x           $end\ for$
xi      $end\ for$
xii  $end\ for$
xiii  $V \leftarrow scramble(v)$
xiv  $return\ V$
```

Pre-processing denotes coefficient-wise multiplications of $v_i$ and $\gamma_{2N}^{-i}$ before NTT. Post-processing denotes final scaling by $N^{-1}$ in the classic NTT$^{-1}$ and coefficient-wise multiplication by $\gamma_{2N}^{-i}$ after the classic NTT$^{-1}$.

### *3.3.2. Comparison of NTT with other polynomial multiplication methods*

An efficient polynomial multiplication algorithm can significantly enhance the efficiency of cryptographic schemes. Several techniques have been developed to improve the efficiency of this operation, including School Book [11], Karatsuba algorithm [12], FFT algorithm, NTT [26], and 1PtNTT [28]. Table 2 shows time complexities of these algorithms for multiplying $n$-degree polynomials.

Table 2: Comparison of NTT with other polynomial multiplication methods

| **Method** | **Time Complexity** | **Multiplication Technique** |
|---|---|---|
| School Book [11] | $O(n^2)$ | Multiplying each term of one polynomial with every term of the other |
| Karatsuba [12] | $O(n^{1.58})$ | Divide-and-conquer technique |
| 1PtNTT [28] | $O(7/4nlogn + 3/2n)$ | NTT that operates on a single point of the polynomials |
| NTT [26] | $nlogn$ | Element-wise multiplication |

School book method is $O(n^2)$ time complex. Karatsuba technique is relatively easy to be comprehended despite the cost is 5/3 times NTT, and 1PtNTT's time complexity would be 7/6 times NTT.
By comparing the time complexities of these methods, it is evident that NTT is the most efficient algorithm for polynomial multiplication. Its time complexity of $n \log n$ makes it an optimal choice for enhancing the efficiency of cryptographic schemes.

## 4. Proposed MLWR-based PKE Scheme using NTT

Present work proposed MLWR-based PKE scheme that uses NWC technique with NTT. The scheme aims to provide a secure and efficient encryption and decryption mechanism based on the MLWR problem. Major components of the proposed scheme are 1) Key generation algorithm, 2) Encryption algorithm, 3) Decryption algorithm, 4)Parameter selection, and 5) Correctness and security of scheme.

### *4.1. Key Generation Algorithm*

Key generation ***algorithm 1***, is used to generate a public key pair $(\delta, b^T)$ and secret key $\mathbf{s}$, which are required for the encryption and decryption processes. Key generation Algorithm generates seed $\delta$ using the Shake-256 hash function. Further, this seed $\delta$ is used to produce a matrix $\mathbf{A}$ of size $k \times k$, where elements of $\mathbf{A}$ are polynomials under the ring $R_q$. Parameters $q$, $p$, and $N$ are selected based on the requirements of NTT and LWR. Modulus $q$ is chosen as a prime number, rounding modulus $p$ is less than $q$, and the degree of the polynomial $N$ is set to 256.
Algorithm generates first part of the key i.e. secret key $\mathbf{s}$ by sampling a short vector $\mathbf{s}$ in the ring $R_q = \mathbb{Z}_q[X]/(X^N + 1)$ with small coefficients. Public key required is $\mathbf{As}$. To speed up matrix-vector multiplication, algorithm performs NTT of both matrix $\mathbf{A}$ and vector $\mathbf{s}$. Multiplication is performed in NTT domain, resulting in the product $\hat{\mathbf{A}}\hat{\mathbf{s}}$. Result is subsequently transformed back to the coefficient form using inverse NTT ($\mathrm{NTT}^{-1}$).
Finally, algorithm computes the public key $b^T$ by taking a rounded value of $\frac{p}{q}$ times the inner product of the secret key $\mathbf{s}$ and matrix $\mathbf{A}$. Rounding operation is performed with respect to $p$, and result is denoted as $\lfloor \frac{p}{q} < \mathbf{s}, \mathbf{A} > \rceil_p$.
The proposed scheme uses NTT for an efficient polynomial multiplication between vector $\mathbf{s}$ and matrix $\mathbf{A}$, which significantly reduces key generation time.
Following pseudocode provides a detailed procedure for key generation algorithm:

**Algorithm 1** KeyGeneration($\delta$)

```
Require: Security parameter $k$
Ensure: Public key $pk$, Secret key $sk$
1: $\delta \leftarrow \{0,1\}^{256}$
2: $\mathbf{s} \leftarrow S_\eta^k \in R_q^{k\times 1}$
3: $\hat{\mathbf{s}} \leftarrow NTT(\mathbf{s})$
4: $\mathbf{A} \in R_q^{k\times k} := Expansion(\delta)$
5: $\hat{\mathbf{A}} \leftarrow NTT(\mathbf{A})$
6: $\mathbf{As} \leftarrow NTT^{-1}(\hat{\mathbf{A}}\hat{\mathbf{s}})$
7: $b^T = \lfloor \frac{p}{q}\mathbf{As} \rceil_p \in R_p^{k\times 1}$
8: $return(pk = (\delta, b^T), sk = (\mathbf{s}))$
```

## *4.2. Encryption Algorithm*

Encryption ***algorithm 2*** takes an $m$-dimensional plaintext vector as input. To ensure the efficient nature of the scheme, algorithm generates a small encrypted text.

Initially, algorithm generates a matrix $\mathbf{A}$ using the seed $\delta$. Then, an $m$-dimensional vector $\mathbf{a}$ is randomly chosen from $\mathbb{Z}_q$. Multiplication of $\mathbf{a}$ with $\mathbf{A}$ is computed using NTT and NTT$^{-1}$ operations, and rounded result is stored in the variable $u$, which is obtained as $\lfloor \frac{p}{q}\mathbf{Aa}\rceil_p$.

Using the public key $b^T$ (from Algorithm 1) and plaintext bit, algorithm computes ciphertext $c$ as a sum of $b^T$ and rounded value of plaintext *bit* multiplied by $\frac{q}{2}$, denoted as $[b^T + \lfloor bit \cdot \frac{q}{2}\rfloor]$.

In summary, encryption algorithm generates a small encrypted text by multiplying random vector $\mathbf{a}$ with matrix $\mathbf{A}$ using NTT operation and then combines it with the public key to obtain ciphertext.

Following pseudo-code provides a detailed step-by-step procedure for proposed encryption algorithm.

**Algorithm 2** LWREncryption $(pk, bit)$

```
Require: public key $pk$, and message $bit$
1: $\mathbf{A} \in R_q^{k\times k} := Expansion(\delta)$
2: $\hat{\mathbf{A}} \leftarrow NTT(\mathbf{A})$
3: $\mathbf{a} \leftarrow S_\eta^k$
4: $\hat{\mathbf{a}} \leftarrow NTT(\mathbf{a})$
5: $\mathbf{Aa} = NTT^{-1}(\hat{\mathbf{A}}\hat{\mathbf{a}})$
6: $u = \lfloor \frac{p}{q}\mathbf{Aa}\rceil_p$
7: $c = [\mathbf{b}^T\mathbf{a} + bit \cdot \lfloor \frac{q}{2}\rfloor]$
8: $return(u, c)$
```

## *4.3. Decryption Algorithm*

Decryption ***algorithm 3*** aims to decrypt ciphertext and recover original plaintext. Decryption process involves the following steps: Initially, algorithm computes a vector $d_1$ by performing a point-wise multiplication of the encrypted text $u$ and secret key $\mathbf{s}$. This multiplication is done using NTT operation by converting $u$ and $\mathbf{s}$ into their NTT forms, denoted as $\hat{u}$ and $\hat{\mathbf{s}}$. Result, $d_1$, is then converted back into coefficient form.

Next, algorithm subtracts $d_1$ from the ciphertext $c$ to obtain decrypted value $d$

To recover original plaintext bit, algorithm checks range of $d$ modulo $q$. If $\frac{-q}{4} \leq d \leq \frac{q}{4}$, plaintext bit is determined to be '0'. If $\frac{q}{4} < d \leq \frac{3q}{4}$, plaintext bit is determined to be '1'. Correctness of scheme is discussed in section 4.5.

All computations are performed under the modulus $q$. Pseudocode provided below outlines detailed procedures of proposed decryption algorithm.

**Algorithm 3** LWRDecryption ($\mathbf{s}, c, u$)

```
Require: Secret key $sk = \mathbf{s}$ and cipher text ($c$ $u$)
1: $\hat{\mathbf{u}} \leftarrow NTT(\mathbf{u})$
2: $\hat{\mathbf{s}} \leftarrow NTT(\mathbf{s})$
3: $\mathbf{us} = NTT^{-1}(\hat{\mathbf{u}}\hat{\mathbf{s}})$
4: $d_1 = \mathbf{us}$
5: $d = c - d_1$
6: $if(((d \ mod \ q) \geq \frac{-q}{4})$ AND $((d \ mod \ q) \leq \frac{q}{4}))$
7: $then$
8:     $d = 0$
9: $elseif(((d \ mod \ q) > \frac{q}{4})$ AND $((d \ mod \ q) \leq \frac{3q}{4}))$
10: $then$
11:     $d = 1$
12: $return \ d$
```

## 4.4. Parameter Selection

Parameter selection is a crucial aspect of MLWR-based PKE scheme. Table 3 presents chosen parameters for the proposed scheme:

Table 3: Parameters for the proposed scheme

| **Parameters** | **Value** |
|---|---|
| Coefficient Modulus | Prime $q = 8380417$ , $p = 4159$ |
| Secret coefficient range $\eta$ | [-2,2] for high level security, [-4,4] for recommended security |
| Polynomial ring | $X^{256} + 1$ |
| Rank of module $k$ | 2 (for security level 1), 3 (for security level 2), 4 (for security level 3), 5 (for security level 4) |
| Multiplication algorithm | NTT |

Most lattice-based PKE schemes are constructed in one of two ways. They either have parameters $k, l > 1$ and use a ring $R_q = \mathbb{Z}_q$, or they are based on the parameters $l = k = 1$ (where $k$ is a rank of a module) with a ring $R_q = \mathbb{Z}_q[X]/(X^N + 1)$. Latter approach yields schemes (e.g., NTRU, BLISS) ingrained in the difficulty of RLWE [13] and Ring-SIS (RSIS) [8] problems. Approaches ingrained in the LWR/SIS/LWE problem use $R_q = \mathbb{Z}_q[X]/(X^N + 1)$ parameter set. There is also an approach based on the MLWR/Module-SIS/MLWE problem [5] for the general case when $l, k \geq 1$ and $R = \mathbb{Z}_q[X]/(X^N + 1)$. Products $kN$ and $lN$ determine the effectiveness of lattice reduction attacks on that scheme, where $N$ denotes a degree of the ring $R_q$. It is generally recommended to set $l, k = 1$ and select a large ring $R_q$ for optimal security.

Value of $N$ has been set as 256 to employ 256 bits of entropy for encrypting the message (plaintext size = 256 bits). Using a small value for $N$ results in reduced noise but leading to lower security. However, capacity to scale security via parameter $k$ is constrained by larger $N$ values.

Our present strategy is to keep a ring $R_q = \mathbb{Z}_q[X]/(X^{256} + 1)$ constant while varying $l$ and $k$. Although this parameter set is suboptimal for the problem scope, using $l = k = 4$ achieves equivalent results compared to using $l = k = 1$ with a ring of dimension 1024.

Matrix $A \in R_q^{k\times l}$ has been chosen to provide flexibility, which is a significant benefit of the scheme. Encryption and decryption algorithms spend a considerable amount of time performing matrix multiplication in a ring $Z_q[X]/(X^{256}+1)$. Thus, improving multiplication component is critical for making the system efficient. Since ring is same for any $k$, $l$, essential components of implementation can be reused for schemes of varying security levels. Changing rings to adjust security in RSIS/RLWR/RLWE-based schemes requires the re-implementation of all cryptographic routines, leading to a significant increase in hardware costs. Proposed approach keeps a ring fixed and modifies parameters $k$ and $l$ to allow a change in a security level. Additionally, scheme offers benefits. First, its less algebraic structure inherently reduces the feasibility of certain practical attacks. Second, it allows for independent variation of parameters $k$ and $l$, providing greater flexibility in security-performance trade-offs.

Selection of a small prime $q$ is necessary to facilitate NTT. To achieve the same, prime $q$ must be chosen such that a group $R_q$ contains an element of order $2N = 512$, or equivalently $q \equiv 1 \pmod{512}$. To satisfy the required condition for an MLWR-based scheme with NTT, a prime modulus $q^*$ is needed, where $q^*/2 > 256 \times 4 \times 2^{12}$. This ensures that modulus is large enough to accommodate the maximum possible coefficient in the product polynomial and avoid precision errors introduced by modular reduction.

Coefficients of the polynomial $\mathbf{s}$ and the public vector of matrix $\mathbf{A}$ are drawn uniformly from $\mathbb{Z}_q$ using rejection sampling [13]. In rejection sampling, random numbers of the desired bit size are generated from the (SHAKE-128) PRNG as candidate samples. Only numbers smaller than $q$ are accepted. Rejection probability refers to the probability that a random number will not be accepted. It can be computed as $(1 - \frac{q}{2^{\log q}})$.

Different primes have varying rejection probabilities, and some may be as large as 50%, which can pose a constraint in lattice-based protocols. However, as shown in Table 4 prime 8380417 has a near-zero rejection probability. This value is approximately equal to $(256 \times 4 \times 2^{12}) - 2^{12}$. By choosing modulus = 8380417, failure rate is nearly zero if the secret polynomial has at least 255 coefficients equal to -4 or 4.

Table 4: Primes and their rejection probabilities

| **Prime** | **Rejection probability** |
|---|---|
| 7681 | 0.06 |
| 12289 | 0.25 |
| 40961 | 0.37 |
| 65537 | 0.50 |
| 120833 | 0.08 |
| 133121 | 0.49 |
| 184321 | 0.30 |
| 8380417 | $\approx 0$ |
| 8058881 | 0.04 |
| 4205569 | 0.50 |
| 4206593 | 0.50 |
| 8404993 | 0.50 |

#### *4.4.1. Parameter comparison with other lattice-based schemes*

A comparison of the parameters of proposed PKE with existing lattice-based PKEs, along with an overview of the parameter settings for each scheme, is provided in Table 5

It has been observed that all schemes, except Frodo, employ polynomials with 256 coefficients and perform polynomial multiplication in the polynomial ring $X^{256} + 1$. A common choice of ring ensures compatibility and facilitates interoperability among different schemes.

However, there is variation in the choice of moduli across the protocols. SABER [3] and Frodo [23] deviate from using prime moduli and instead utilize power-of-2 moduli. This deviation poses challenges for a straightforward application of the NTT, a technique commonly used for efficient polynomial multiplication. To overcome above limitation and save computing time, proposed scheme constructs a public matrix $\mathbf{A}$ in NTT form. Chosen parameters enable proposed MLWR scheme to use NTT modulo $q$ for polynomial multiplication, ensuring efficient and secure computation. Table 5 summarizes the specific parameter values employed by each scheme. Coefficient modulus $q$ differs for each scheme, with values such as 7681, $2^{10}$, $2^{16}$, and 8380417 in the proposed scheme. Secret coefficients, which determine the range of possible values for private keys, also vary among the protocols.

These comparisons highlight distinct parameter choices made in each scheme and emphasize adaptability and efficiency of proposed MLWR-based scheme, which incorporates NTT for polynomial multiplication while employing carefully selected moduli. By leveraging these parameter settings, proposed scheme achieves a balance between security, efficiency, and compatibility with existing lattice-based PKE schemes.

Table 5: Comparison of proposed scheme's parameters with existing lattice-based PKE schemes

| **Parameter** | **Kyber [5]** | **Saber [3]** | **Frodo [23]** | **Proposed Scheme** |
|---|---|---|---|---|
| Coefficient Modulus $q$ | 7681 | $2^{13}$ | $2^{16}$ | 8380417 |
| Secret coefficients range for Low level security | $[-5, 5]$ | $[-5, 5]$ | $[-6, 6]$ | $[-4, 4]$ |
| Secret coefficients range for Recommended level security | $[-5, 5]$ | $[-4, 4]$ | $[-6, 6]$ | $[-2, 2]$ |
| Secret coefficients range for High level security | $[-3, 3]$ | $[-3, 3]$ | $[-6, 6]$ | $[-2, 2]$ |
| Degree N | 256 | 256 | 640 | 256 |

### *4.5. Correctness of the proposed MLWR-based PKE*

Correctness property ensures that proposed scheme operates accurately and yields the intended results under certain conditions. To get accurate results, the value of rounding errors must be bounded. Following Theorem 4.1 provides the formal statement of correctness for proposed scheme.

**Theorem 4.1.** *Proposed scheme will work correctly if the following condition holds:*

*$Pr \parallel e_r \parallel$ is in the range $[\frac{-q}{4}, \frac{q}{4}]$ is high, where $e_r$ is a rounding error.*

*Proof.* Let us consider ciphertext preamble $u = \lfloor \frac{p}{q}\mathbf{Aa}\rceil_p$,and ciphertext payload $c = [\mathbf{b}^T\mathbf{a} + bit \cdot \lfloor \frac{q}{2} \rfloor]$. If $(u, c)$ is correctly formulated ciphertext then decrypted value $d$ is $c - d_1 = c - \mathbf{s}^T u$

Expanding the expression, we have:

$$c - \mathbf{s}^T u = b^T\mathbf{a} + bit \cdot \lfloor \frac{q}{2} \rfloor - \lfloor \frac{p}{q}\mathbf{Aa}\rceil_p \tag{2}$$

In the key generation pseudo code (refer section 4.1), it is stated that $b^T = \lfloor \frac{p}{q}\mathbf{s}^T\mathbf{A}\rceil_p$

$$\text{and} \qquad \lfloor \frac{p}{q}\mathbf{s}^T\mathbf{A}\rceil_p \approxeq \mathbf{s}^T\mathbf{A} + e_r$$

As $p/q$ would introduce small deterministic error in $\lfloor \frac{p}{q}\mathbf{s}^T\mathbf{A}\rceil_p$, where $p < q$, and $e_r$ is a rounding error.

$$\therefore \qquad b^T \approxeq \mathbf{s}^T\mathbf{A} + e_r$$

$$\text{and} \qquad \lfloor \frac{p}{q}\mathbf{Aa}\rceil_p \approxeq \mathbf{Aa} + e_r$$

by substituting value of $b^T$ and $\lfloor \frac{p}{q}\mathbf{Aa}\rceil_p$ in equation 2

$$\therefore \qquad c - \mathbf{s}^T u = (\mathbf{s}^T\mathbf{A} + e_r)\mathbf{a} + bit \cdot \lfloor \frac{q}{2} \rfloor - \mathbf{s}^T(\mathbf{Aa} + e_r)$$

$$\therefore \qquad c - \mathbf{s}^T u = \mathbf{s}^T\mathbf{Aa} + e_r\mathbf{a} + bit \cdot \lfloor \frac{q}{2} \rfloor - \mathbf{s}^T\mathbf{Aa} + \mathbf{s}^T e_r$$

$$\text{or} \qquad c - \mathbf{s}^T u = e_r(\mathbf{a} + \mathbf{s}^T) + bit \cdot \lfloor \frac{q}{2} \rfloor \tag{3}$$

In Equation 3, it is evident that decrypted value depends on the value of the rounding error $e_r$. Notably, vectors **a** and **s** are significantly smaller compared to the rounding error $e_r$. To ensure correctness, it is crucial to select threshold value $\beta$ such that the probability of $\| e_r \| \in \beta$ is high. Consequently, value $\| e_r(\mathbf{a}+\mathbf{s}^T) \|$ will fall within the range $bit \cdot \beta$. The decryption algorithm guarantees correctness when $\| e_r(\mathbf{a}+\mathbf{s}^T) \| \in [\frac{-q}{4}, \frac{q}{4}]$. It is due to our selection of threshold $\beta$ for $e_r$ as $[\frac{-q}{4} \cdot bit, \frac{q}{4} \cdot bit]$. With mentioned parameter selection, the condition $\| e_r(\mathbf{a}+\mathbf{s}^T) \| \in [\frac{-q}{4}, \frac{q}{4}]$ holds with a high probability, ensuring correctness of proposed MLWR-based PKE scheme. □

## *4.6. Security*

Security of proposed MLWR-based PKE scheme can be established under the hardness assumption of MLWR problems. Following Theorem 4.2 formally proves the security argument.

**Theorem 4.2.** *Proposed PKE is indistinguishable chosen-plaintext (IND-CPA) secure under the hardness assumption of $MLWR_{k,N+l,q,p}(U_0)$.*

*Proof.* To prove the IND-CPA security, it needs to be demonstrated that the public information $pk = (\mathbf{A}\|b^T) \leftarrow$ KeyGeneration($\delta$) is computationally indistinguishable from the uniform distribution $\mathbb{Z}_q^{k\times(l\times N)}$.
Let $U_0$ and $U_1$ be two distributions defined as follows:

- $U_0 = (pk = (\delta, b^T), (u, c)) : pk \leftarrow KeyGeneration(\delta), c \leftarrow Z_q^{l\times N}$
- $U_1 = (pk = (\delta, b^T), (u, c)) : pk \leftarrow Z_q^{k\times(l\times N)}, c \leftarrow Z_q^{l\times N}$

Consider $U$ be a uniform distribution. Public key $pk = (A \| b^T) \leftarrow KeyGeneration(\delta)$ is generated by sampling $k$ instances of MLWR sample problem with independent secret vectors $s_1, s_2, \cdots, s_k \leftarrow U$. Moreover, according to [10], MLWR problem is no easier than ordinary LWR problem. Therefore, distributions $U_0$ and $U_1$ are computationally indistinguishable under the $MLWR_{k,N+l,q,p}$ assumption (refer section 3.2).
Thus, proposed PKE scheme achieves indistinguishable chosen-plaintext security under the hardness assumption of $MLWR_{k,N+l,q,p}(U_0)$. □

## *4.7. Implementation Details*

Implementation details of the proposed PKE scheme are as follows:
Initial step in the implementation is a generation of the matrix $\mathbf{A} \in R_q^{k\times k}$ from a random seed $\delta \leftarrow 0, 1^{256}$. In present work, SHAKE-128 hash function is used to expand the seed into matrix **A**. The output stream derived from SHAKE-128 is treated as a series of integers within the range of 0 to $2^{23} - 1$. This transformation is achieved by clearing the highest bit of every third byte and interpreting blocks of three consecutive bytes in little-endian byte order. In the next step, expansion employs rejection sampling on these 23-bit integers to select and sample the 256 coefficients. Expansion function ensures that **A** is uniformly sampled and provides sufficient randomness with high probability. Matrix **A** is used both for key generation and multiplication operations in the encryption process. To make the multiplication

faster, NTT algorithm is used. Matrix $\mathbf{A}$ needs to be uniformly sampled and expansion function takes the seed as input and outputs a uniform matrix $\mathbf{A}$ in NTT format with coefficients in a range of 0 to $q - 1$.

***Expansion function:***

Expansion function maps a 256-bit seed $\delta$ to the values of a uniform matrix $\mathbf{A}$ in its NTT representation. Function absorbs 32 bytes of the seed to compute coefficients of the matrix. Process of expanding the seed $\delta$ into a polynomial matrix $\mathbf{A}$ is a significant and time-consuming operation. Since the matrix $\mathbf{A}$ is required for encryption and key generation, implementation of SHAKE-128 is crucial for efficiency of the scheme.

Matrix $\mathbf{A} \in R_q^{k \times k}$ is uniformly random, where $k = 4$, and elements are polynomials under a ring $R_q = \mathbb{Z}_q[X]/(X^N + 1)$. Here, $N = 256$ represents a degree of the polynomial ring, and $k$ is rank of the module, which determines dimension of the fundamental lattice problem $(N \cdot k)$.

After generation of matrix $\mathbf{A}$, next step is to sample a short vector $\mathbf{s}$ using rejection sampling [13]. This short vector serves as a secret key.

Public key $pk$ consists of two parts (as explained in Section 4.1): First part is a seed $\delta$, and second part is $b^T = \lfloor \frac{p}{q} \mathbf{A}\mathbf{s} \rceil_p$. Matrix multiplication is a crucial operation in this scheme. Present work uses NTT form of $\mathbf{A}$ and $\mathbf{s}$ to perform pointwise multiplication. Result of this multiplication is then subjected to the inverse NTT transform and multiplied by the Montgomery factor [29] for modular reduction. Montgomery algorithm [29] is employed for modular reduction. Rounding operation is applied to $b^T$ to reduce the size of a public key. In LWR scheme, parameter $p$ acts as a rounding modulus, allowing public key $pk$ to be represented in $R_p$ instead of $R_q$, where $q$ is greater than $p$.

The additional implementation details are as follows:

- **Sampling secret vector s**

  Expansion function maps a seed $\delta$ to the secret vector $\mathbf{s} \in S_\eta^k$. Each of the $k$ polynomials in $\mathbf{s}$ is evaluated individually. For $i^{th}$ polynomial, where $0 \leq i \leq k$, the function absorbs 64 bytes of $\delta$ concatenated with 2 bytes representing $k + i$ in little-endian byte order into SHAKE-256. Output of SHAKE-256 is used to generate a sequence of uniformly random positive values in the range $0, 1, ..., 2\eta - 1, 2\eta$ using reject sampling. Each output byte is divided into two numbers in the range $0, 1, ..., 14, 15$, using lower and upper four bits. For $\eta = 2$, an integer is considered acceptable if it is below 15; if it exceeds 15, it is reduced modulo 5. Finally, polynomial coefficients are derived by subtracting the positive values from $\eta$.

- **Polynomial Multiplication**

  Main algebraic operation in the scheme involves a multiplication of a matrix $\mathbf{A}$, whose components are polynomials in $\mathbb{Z}_q[X]/(X^{256} + 1)$, with a vector $\mathbf{s}$ containing polynomials in the same ring. High-security parameter setup uses a $4 \times 4$ matrix $\mathbf{A}$ composed of 16 polynomials. Therefore, multiplication $\mathbf{As}$ requires 16 polynomial multiplications. To accelerate these multiplications, NTT is used. NTT is a version of the FFT algorithm that operates over a finite field $\mathbb{Z}_q$ instead of the complex numbers. In order to utilize "fully-splitting" NTT method, a prime number $q$ is chosen such that a group $\mathbb{Z}_q^*$ has an element of rank $2N = 512$ (or $q \equiv 1 \mod 512$). NTT allows representation of polynomials in the form of CRT [25] coefficients, which simplifies polynomial multiplication through coordinate-wise products. Proposed method minimizes computational cost by using the least number of NTT and inverse NTT operations.

Benefit of using NTT format is that significantly fewer NTT calculations are required when performing matrix-vector multiplication $\mathbf{As}$ compared to the dimensions of a matrix. Since a matrix is always random, it can be precomputed in the NTT format and stored. Multiplication of a matrix $\mathbf{A}$ with a vector $\mathbf{s} \in R_q^4$ using NTT involves four NTT calculations to convert $\mathbf{s}$ into the NTT format, followed by point-wise multiplication to obtain product $\mathbf{As}$ in the NTT representation. Finally, four inverse NTT operations are performed to obtain a final result.

In the proposed NTT implementation, integer arithmetic is used instead of floating-point arithmetic. By packing four coefficients into a 256-bit vector register, similar to the density used in floating-point implementations, multiplication speed can be improved by a factor of two. This speedup is achieved by carefully arranging the operations, pipelining the multiplications, and utilizing parallelism throughout the NTT process to hide sections of multiplication latencies.

## 5. Result and Analysis

### *5.1. Experimental setup*

Proposed PKE scheme has been implemented, compiled, and tested on a Linux operating system using the CDAC-PARAM Shavak machine, which is equipped with an Intel(R) Xeon(R) Gold 6226R v4 series CPU, and the program was compiled using the GCC compiler.

### *5.2. Performance analysis*

Execution time and CPU cycles for key generation, encryption, and decryption functions were recorded using the Intel VTune Profiler tool. Results are shown in Table 6.

CPU time utilized was 103.448 $\mu s$ for the execution of key generation function, 75 $\mu s$ for encryption, and 34.483 $\mu s$ for decryption. Approximate CPU cycles required to execute each function were 400K, 320K, and 150K respectively. Computed result offers valuable information regarding the performance of proposed scheme on the specific hardware platform.

Table 6: Execution time and CPU cycles for proposed scheme

| **Key Generation**[1] | **Encryption**[1] | **Decryption**[1] |
|---|---|---|
| Avg. Execution Time in ($\mu s$) | | |
| 103.448 | 75 | 34.483 |
| Approx. CPU Cycles (Thousands) | | |
| 400 | 320 | 150 |

[1] Average CPU time/cycles w.r.t. execution on Intel(R) Xeon(R) Gold 6226R v4 series with CISC

### 5.2.1. *Performance with different parameter sets*

Advantage of Module LWR is that a same polynomial ring $R_q$ is used for all levels of security [24] from weak to high. To achieve a stronger security level, the scheme uses a module with a larger rank. Table 7 shows proposed scheme's performance with different parameter sets and respective security. Parameter sets, denoted as $P_1$, $P_2$, $P_3$, and $P_4$, correspond to increasing security levels, ranging from weak to high.

Table 7: Performance of proposed scheme with different parameter sets

| **Security level and Parameter set** | **Operation** | **Avg. CPU time ($\mu$s )** | **Avg. CPU cycles** |
|---|---|---|---|
| $P_1$ - Level 1-Weak : $\eta$=4, k=2, l=2 | Key Generation | 68.965 | 200,000 |
| | Encryption | 34.483 | 100,000 |
| | Decryption | 24.483 | 100,000 |
| $P_2$ - Level 2 - Medium : $\eta$=2, k=3, l=3 | Key Generation | 68.966 | 250,000 |
| | Encryption | 49.483 | 150,000 |
| | Decryption | 34.483 | 120,000 |
| $P_3$- Level 3-Recommended: $\eta$=4, k=4, l=4 | Key Generation | 103.488 | 400,000 |
| | Encryption | 75 | 320,000 |
| | Decryption | 34.483 | 150,000 |
| $P_4$ - Level 4-High : $\eta$=4, k=5, l=5 | Key Generation | 206.896 | 600,000 |
| | Encryption | 103.448 | 400,000 |
| | Decryption | 38.483 | 200,000 |

These results demonstrate the trade-off between security level and performance. As the security level increases, computational cost (in terms of CPU time and cycles) also increases. It is important to select an appropriate parameter set based on the desired security level.

## *5.3. Security estimation*

A security estimation of the proposed MLWR-based PKE scheme was conducted for various parameter sets. Security evaluation is based on the estimation of computational costs required to solve associated hard problems using Block Korkine-Zolotarev (BKZ) [30] attack. BKZ attack is used to solve exact SVP as a subroutine in smaller dimensions and plays a significant role in analyzing security of lattice-based cryptographic schemes. Main idea behind BKZ is to transform an initial basis of a lattice into a shorter and more orthogonal basis, which facilitates finding shorter lattice vectors and makes lattice problems easier to solve. Attack operates in multiple block steps ($\beta$), and in each step, it reduces the length of certain vectors while preserving the properties of the lattice. Process repeats until a suitable basis is achieved.

The lattice estimator tool [30], maintained by Martin Albrecht, is used for security estimation purpose. Lattice estimator tool is widely used in lattice-based cryptography to detect security of schemes including SABER [3] and Kyber [5]. Estimator provides a convenient way to assess the security of approach with proposed parameters against known attacks. Parameters w.r.t. proposed MLWR-based PKE scheme are provided to the lattice estimator tool, including the lattice dimension ($n$), modulus ($q$), secret distribution method, and specific hard problem used. Estimator execution provides estimated costs for solving the hard problems associated with the scheme.

Table 8 provides a summary of the security estimations for proposed MLWR-based PKE scheme against a BKZ attack. It includes different parameter sets with specific lattice dimensions, moduli, delta values, and corresponding block sizes.

Table 8: Security estimation of proposed scheme for BKZ

| **Security Level** | **Proposed parameter set** | **Block size $\beta$** | **Required security defined at various levels by NIST [24]** | **Estimated ≈Cost of proposed scheme** |
|---|---|---|---|---|
| Low | $n = 512, q = 8380417, \delta = 1.41$ | 6 | $\geq 2^{128}$ | $2^{154.4}$ |
| Medium | $n = 768, q = 8380417, \delta = 1.0$ | 8 | $\geq 2^{192}$ | $2^{201.1}$ |
| Recommended | $n = 1024, q = 8380417, \delta = 1.41$ | 10 | $\geq 2^{256}$ | $2^{270.9}$ |
| High | $n = 1280, q = 8380417, \delta = 1.41$ | 13 | – | $2^{317.3}$ |

Note: Estimated cost represents an approximation of the computational effort required to solve a hard problem.

Proposed MLWR-based PKE scheme demonstrates robust security against the BKZ attack for all three parameter sets. Security level increases with larger dimension ($n$) and is also influenced by delta ($\delta$) parameter. Above estimations provide valuable insights into the strength of the scheme and its resistance to lattice-based BKZ attack.

Table 8 shows that proposed scheme (specifically, parameter sets for recommended and high security) provides security greater than $2^{256}$ for BKZ meeting the requirements for NIST security categories 5 [24]. By carefully selecting

parameters and relying on the hardness of lattice problems, proposed scheme showcases strong security against lattice-based attacks. Consequently, it holds promise as a viable solution for post-quantum cryptography applications.
Proposed MLWR-based PKE scheme emphasizes efficient implementation while meeting the recommended security requirements. Scheme uses NTT method to improve efficiency of the underlying polynomial multiplication, thereby accelerating key generation, encryption, and decryption processes.
Combination of robust security against BKZ attack, adherence to recommended security levels, and efficient implementation using NTT make proposed MLWR-based PKE scheme a promising solution for post-quantum cryptography.

## 6. Comparison with the other lattice-based schemes and analysis

Proposed scheme's performance is compared with other lattice-based schemes, namely Kyber, Saber, and Frodo, which are efficient candidates of the NIST-PQC project's second-round finalists [2]. The comparison focuses on the recommended security level of NIST.

### *6.1. Comparison of polynomial multiplication method used in proposed PKE with other lattice-based schemes*

NTT-like polynomial multiplication is not implemented natively in SABER due to the use of a two-power modulus. Saber and existing MLWR-based schemes employ asymptotically more slow polynomial multiplication techniques like Schoolbook [11], Karatsuba [12], ToomCook, or combinations of these. Table 9 shows that proposed polynomial multiplication results (for parameter set p3 and dimension of the lattice = 1024) need fewer cycles as well as less execution time than [3, 23, 5].
Simulations and experiments were carried out in Secure Systems Lab, CSE Department, SoCE&MS, DIAT, Pune on the CDAC-PARAM Shavak machine (Intel(R) Xeon(R) Gold 6226R v4 series with CISC).

Table 9: Comparison of polynomial multiplication method in proposed PKE with other PKE schemes

| **Method** | **Used polynomial multiplication method** | **CPU cycles** | **Avg. execution time ($\mu$ sec.)** |
|---|---|---|---|
| MLWR [3] | Karatsuba and ToomCook | 5,600,000 | 1379 |
| LWE [23] | Naive (schoolbook) | 1,699,100,000 | 250138 |
| MLWE [5] | NTT | 5557,900,000 | 61310 |
| Proposed scheme - MLWR | NTT | 300,000 | 793 |

### *6.2. Comparison of performance of proposed PKE with other lattice-based PKE schemes*

Performance of the proposed PKE has been compared with other lattice-based schemes w.r.t. keys size and required execution time for encryption and decryption process.
Table 10 presents various key sizes of proposed scheme and other lattice-based schemes for recommended security.

Public key and ciphertext sizes of the proposed scheme are smaller than the Frodo scheme. Secret key size is smaller than Kyber, SABER, and Frodo schemes.

Table 10: Comparison of proposed scheme's key sizes (bytes) with existing lattice-based PKE schemes

| **Feature** | **Kyber [5]** | **Saber [3]** | **Frodo [23]** | **Proposed Scheme** |
|---|---|---|---|---|
| Secret key(sk) size | 1,632 | 1,568 | 19,888 | 1,422 |
| Public key(pk) size | 800 | 672 | 9,616 | 1,040 |
| Ciphertext(ct) size | 736 | 736 | 9,720 | 2,647 |

Although a public key and ciphertext sizes are larger than [3, 5], the software implementation of the proposed scheme achieved speedy execution by using NTT. As presented in Figure 1, average execution time of a encryption and decryption process of proposed scheme is comparable with the most efficient scheme SABER which was implemented in hardware using FPGA. Average time required to encrypt a 256-bytes message is approximately equal to SABER, whereas the average decryption time is less than SABER. Since the best software implementations will never surpass the best hardware implementations therefore the hardware implementation of the proposed technique may be quicker than current implementation.

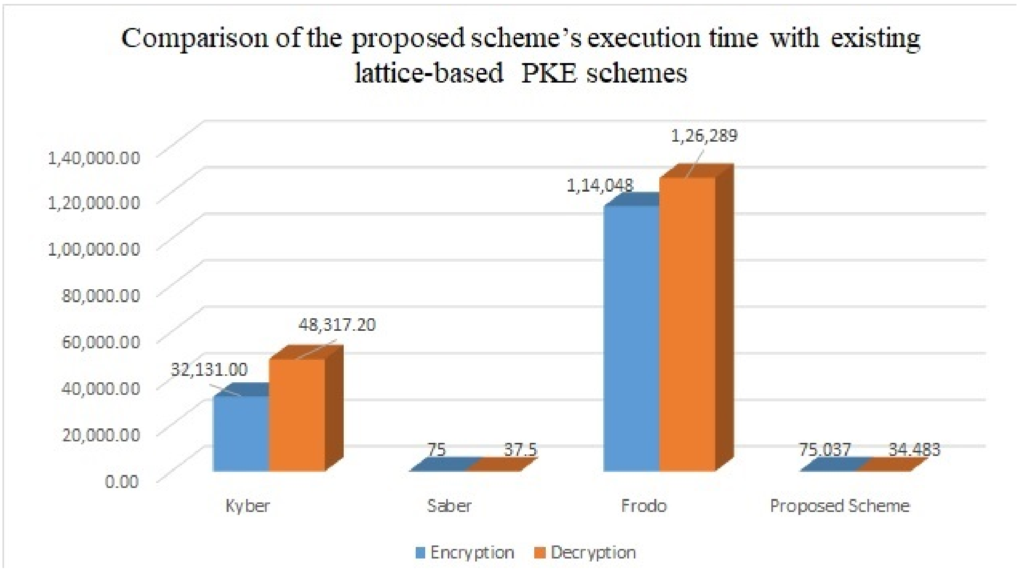

Figure 1: Comparison of proposed scheme's execution time ($\mu s$) with existing lattice-based PKEs

## *6.3. Comparison of features of proposed PKE with other lattice-based schemes*

The advantage of the proposed PKE over the other PKEs is it adheres to recommended security practices, ensuring efficiency. Additionally correctness proof, performance and security analysis of the proposed PKE scheme have been reported. Table 11 provides a comprehensive comparison between the proposed PKE scheme and other relevant approaches. The table compares these schemes based on various features, including the underlying hard problem, chosen parameters, polynomial multiplication method, correctness proofs, performance analyses, security analysis details, and other minor features.

Table 11: Comparison of proposed PKE with other lattice based schemes

| **Features** | **Kyber [5]** | **Saber [3]** | **Frodo [23]** | **Proposed Scheme** |
|---|---|---|---|---|
| Underlying hard problem | MLWE | MLWR | LWE | MLWR |
| Polynomial space | $\mathbb{Z}_q[X]/(X^{256}+1)$ | $\mathbb{Z}_q[X]/(X^{256}+1)$ | $\mathbb{Z}_q[X]$ | $\mathbb{Z}_q[X]/(X^{256}+1)$ |
| Degree N | 256 | 256 | 640 | 256 |
| Modulus | Prime: 3329 | Power of 2 : 8192 | Power of 2: 32768 | Prime:8380417 |
| Analysis of polynomial multiplication methods | - | - | - | ✓ |
| Polynomial multiplication method used | NTT | Karatsuba and ToomCook | Naive (school-book) | NTT |
| Sampling | Rejection Sampling | No need of rejection sampling | Inversion sampling | Rejection Sampling |
| Secret distribution | Centered binomial distribution | Centered binomial distribution | Uniform distribution | Centered binomial distribution |
| Correctness proof | - | ✓ | ✓ | ✓ |
| Security analysis | ✓ | ✓ | ✓ | ✓ |
| IND-CPA security | ✓ | ✓ | ✓ | ✓ |
| Correctness proof | - | ✓ | ✓ | ✓ |
| Performance analysis | ✓ | ✓ | ✓ | ✓ |
| Key sizes [bytes] | sk:1632, pk:800 | sk:1568, pk:672 | sk:19888, pk:9616 | sk:1422, pk:1040 |
| Avg. Execution time ($\mu$ s) | Enc:32131 Dec:48317.20 | Enc:75, Dec:37.5 | Enc:114038 Dec:126289 | Enc:75.037 Dec:34.483 |
| Avg. CPU cycles required | Enc:6870, Dec:962 | Enc: 200 , Dec:100 | Enc:1480 Dec:1597 | Enc: 320 , Dec:200 |
| BKZ security | $2^{238.3}$ | $2^{292}$ | $2^{241.1}$ | $2^{270.9}$ |

[1] Note: ✓denotes "performed" and - denotes there is no result reported in the particular reference.

Table 11 shows a comprehensive analysis of proposed scheme, encompassing both security and performance. This thorough examination validates the scheme's implementation and security properties. While employing rejection sampling, the scheme maintains efficiency comparable to Saber. It achieves the recommended security level ($\geq 2^{256}$) with compact key sizes.

## 7. Conclusion

Lattice-based cryptography is a promising candidate for post-quantum secure communication. It provides efficient encryption techniques and a wide range of hard problems. However, there is room for improvement in the efficiency of core components, particularly in polynomial multiplication.
Proposed scheme is an efficient MLWR-based PKE scheme which utilises NTT for polynomial multiplication. Scheme achieves significant performance gains compared to existing lattice-based schemes, offering a 2x speedup in polynomial multiplication compared to the Karatsuba-Toom-Cook combination. Proposed scheme demonstrates strong security as it is based on an improved variant of the LWR problem. It boasts of faster and simpler encryption and decryption processes while maintaining compact key sizes, particularly for the secret key (1,422 bytes). Significantly, the size of the secret key is less in comparison to the most efficient lattice-based PKE schemes. Since the proposed scheme is based on a hard problem in lattices, it is resistant to quantum attacks and achieves recommended security levels for NIST categories. The security is demonstrated by the proof provided under the IND-CPA security and security estimations against the BKZ attack using Martin Albrecht's estimator.
Paper presents a detailed performance and security analysis of the proposed scheme. It also gives a comparative analysis with other lattice-based PKEs to highlight its advantages.
Future work in this area includes exploring parallelization of NTT techniques to further enhance the performance of the polynomial multiplication procedure. Optimization techniques can also be investigated to further reduce size of public keys. Additionally, there is potential to apply proposed NTT-based methodologies to a lattice-based signature scheme, expanding the scope of lattice-based cryptography.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## Data Availability Statement

Implemented code and data related to results are available with authors.

## Acknowledgments

From the beginning of the research, we are grateful to Mr. Atul Kumar, Prof. Ajay Mishra, Ms. Lucy Sharma, Dr. Nilesh Ramesh Kolhalkar, and other colleagues for their productive technical conversations, guidance, continuous motivation and support.

## References

[1] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Journal on Computing 26 (5) (1997) 1484–1509. arXiv:9508027, doi:10.1137/S0097539795293172.

[2] G. Alagic, D. Cooper, Q. Dang, T. Dang, J. M. Kelsey, J. Lichtinger, Y.-K. Liu, C. A. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, D. Smith-Tone, D. Apon, Status report on the third round of the nist post-quantum cryptography standardization process (2022-07-05 04:07:00 2022). doi:https://doi.org/10.6028/NIST.IR.8413.
URL https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=934458

[3] M. V. Beirendonck, J.-P. D'anvers, A. Karmakar, J. Balasch, I. Verbauwhede, A side-channel-resistant implementation of saber, ACM Journal on Emerging Technologies in Computing Systems (JETC) 17 (2) (apr 2021). doi:10.1145/3429983.
URL https://doi.org/10.1145/3429983

[4] C. Chen, O. Danba, A. Hülsing, J. Rijneveld, J. M. Schanck, P. Schwabe, W. Whyte, Z. Zhang, NTRU Algorithm Specifications And Supporting Documentation, Brown University and Onboard security company, Wilmington USA (2019) 1–32.

[5] T. L. Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, D. S. Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, CRYSTALS-Kyber Algorithm Specifications and Supporting Documentation, NIST PQC Round 2 (4) (2021) 1–32.
URL https://pq-crystals.org/dilithium/

[6] A. Banerjee, C. Peikert, A. Rosen, Pseudorandom functions and lattices, Annual International Conference on the Theory and Applications of Cryptographic Techniques 7237 LNCS (2012) 719–737. doi:10.1007/978-3-642-29011-4_42.

[7] O. Regev, The learning with errors problem, Proceedings of the Annual IEEE Conference on Computational Complexity 3 (015848) (2010) 191–204. doi:10.1109/CCC.2010.26.

[8] Y. Wang, Y. Zhao, M. Wang, On the Hardness of Ring/Module/Polynomial LWR Problems, Cryptology ePrint Archive (2021).

[9] A. A. Pandit, A. Kumar, A. Mishra, Lwr-based quantum-safe pseudo-random number generator, Journal of Information Security and Applications 73 (2023) 103431. doi:https://doi.org/10.1016/j.jisa.2023.103431.
URL https://www.sciencedirect.com/science/article/pii/S2214212623000169

[10] A. Langlois, D. Stehlé, C. C. by Padro A Langlois, D. Stehlé, A. Langlois, Worst-case to average-case reductions for module lattices, Designs, Codes and Cryptography 2014 75:3 75 (3) (2014) 565–599. doi:10.1007/S10623-014-9938-4.

[11] W. Liu, S. Fan, A. Khalid, C. Rafferty, M. O'Neill, Optimized schoolbook polynomial multiplication for compact lattice-based cryptography on fpga, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 27 (10) (2019) 2459–2463. doi:10.1109/TVLSI.2019.2922999.

[12] A. Weimerskirch, Karatsuba algorithm, in: Encyclopedia of Cryptography and Security, Springer US, Boston, MA, 2011, pp. 666–669. doi:10.1007/978-1-4419-5906-5_35.
URL https://doi.org/10.1007/978-1-4419-5906-5_35

[13] U. Banerjee, T. S. Ukyab, A. P. Chandrakasan, Sapphire: A Configurable Crypto-Processor for Post-Quantum Lattice-based Protocols (Extended Version), Cryptology ePrint Archive 2019 (4) (2019) 17–61. arXiv:1910.07557, doi:10.13154/TCHES.V2019.I4.17-61.

[14] V. Lyubashevsky, D. Micciancio, C. Peikert, A. Rosen, Swifft: A modest proposal for fft hashing, in: K. Nyberg (Ed.), Fast Software Encryption, Lecture Notes in Computer Science, Vol. 5086, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 54–72.

[15] M. Dworkin, Sha-3 standard: Permutation-based hash and extendable-output functions, Tech. rep., Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD (2015-08-04 2015). doi:https://doi.org/10.6028/NIST.FIPS.202.

[16] J. Alwen, S. Krenn, K. Pietrzak, D. Wichs, Learning with rounding, revisited, in: R. Canetti, J. A. Garay (Eds.), Advances in Cryptology – CRYPTO 2013, Lecture Notes in Computer Science, Vol. 8042, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 57–74.

[17] D. Micciancio, Shortest vector problem, in: Encyclopedia of Cryptography and Security, Springer US, Boston, MA, 2005, pp. 569–570. doi:10.1007/0-387-23483-7_392.
URL https://doi.org/10.1007/0-387-23483-7_392

[18] D. Micciancio, S. Goldwasser, Complexity of lattice problems, Vol. 671 of The Springer International Series in Engineering and Computer Science, Springer, New York, NY, 2002.

[19] D. Micciancio, O. Regev, Lattice-based cryptography, in: Post-Quantum Cryptography, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 147–191. doi:10.1007/978-3-540-88702-7_5.

[20] P. Nguyen, J. Stern, Cryptanalysis of the ajtai-dwork cryptosystem, in: Annual International Cryptology Conference, Springer, 1998, pp. 223–242.

[21] J. Hoffstein, J. Pipher, J. H. Silverman, Ntru: A ring-based public key cryptosystem, in: International algorithmic number theory symposium, Springer, 1998, pp. 267–288.

[22] R. Lindner, C. Peikert, Better key sizes (and attacks) for lwe-based encryption, in: Cryptographers' Track at the RSA Conference, Springer, 2011, pp. 319–339.

[23] J. W. Bos, O. Bronchain, F. Custers, J. Renes, D. Verbakel, C. van Vredendaal, Enabling frodokem on embedded devices, Cryptology ePrint Archive, Paper 2023/158, https://eprint.iacr.org/2023/158 (2023).
URL https://eprint.iacr.org/2023/158

[24] NIST, Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process (2016).
URL https://csrc.nist.gov/pqcrypto

[25] M. Cenk, F. Ozbudak, Improved polynomial multiplication formulas over $if_2$ using chinese remainder theorem, IEEE Transactions on Computers 58 (4) (2009) 572–576. doi:10.1109/TC.2008.207.

[26] C.-M. M. Chung, V. Hwang, M. J. Kannwischer, G. Seiler, C.-J. Shih, B.-Y. Yang, NTT multiplication for NTT-unfriendly rings: New speed records for Saber and NTRU on Cortex-M4 and AVX2, IACR Transactions on Cryptographic Hardware and Embedded Systems 2021 (2) (2021) 159–188, artifact available at https://artifacts.iacr.org/tches/2021/a7. doi:10.46586/tches.v2021.i2.159-188.

[27] N. Zhang, B. Yang, C. Chen, S. Yin, S. Wei, L. Liu, Highly efficient architecture of newhope-nist on fpga using low-complexity ntt/intt, IACR Transactions on Cryptographic Hardware and Embedded Systems 2020 (2) (2020) 49–72.

[28] B. Wang, H. Gao, F. Yang, Fast Implementation of Multiplication on Polynomial Rings, Security and Communication Networks 2022 (2022) 4649158. doi:10.1155/2022/4649158.
URL https://doi.org/10.1155/2022/4649158

[29] M. Safieh, J. Freudenberger, Montgomery reduction for gaussian integers, Cryptography 5 (1) (2021). doi:10.3390/cryptography5010006.
URL https://www.mdpi.com/2410-387X/5/1/6

[30] M. R. Albrecht, R. Player, S. Scott, On the concrete hardness of learning with errors, Journal of Mathematical Cryptology 9 (3) (2015) 169–203.