Vol: 2, Issue: 2, 2024
Pages:96-107
Received: 15 October 2024
Accepted: 13 December 2024

CURRENT TRENDS IN COMPUTING
Research Article
DOI: 10.71074/CTC.1562363

ON THE POLYNOMIAL MULTIPLICATION ALGORITHMS FOR POST-QUANTUM CRYPTOGRAPHY

EBRU YALÇIN[1*], FIDAN NURIYEVA[2,3] AND ERDEM ALKIM[2]

[1] *The Graduate School of Natural and Applied Sciences, Department of Computer Science, Dokuz Eylul University, 35390, Izmir, Türkiye*
[2] *Department of Computer Science, Dokuz Eylul University, 35390, Izmir, Türkiye*
[3] *Institute of Control Systems, The Ministry of Science and Education of the Republic of Azerbaijan, Baku, Azerbaijan*

ABSTRACT. **This study explores the multiplication operations carried out on polynomial rings within lattice-based systems used in post-quantum cryptography. Polynomial rings of high degree are utilized to enhance security in post-quantum cryptography. Since multiplication is the most time-consuming arithmetic operation on polynomial rings, several algorithms have been suggested to optimize newly developed systems by enhancing their efficiency. Typically, these algorithms use the properties of the chosen polynomial ring to minimize the number of multiplications, however, some arithmetical tricks can be used to use them for other rings. Therefore, the systems are optimized in terms of efficiency and cost. In this study, we investigated several multiplication algorithms based on their complexity and reported the results from the literature for their implementation efficiency. We have compared those algorithms when they were implemented to perform multiplications on the same polynomial ring and reported that the ring of the coefficients should be also considered when comparing the efficiency.**

1. INTRODUCTION

In the modern day, as technology advances and becomes more widely utilized, the need to guarantee the security of systems and networks has become a significant concern due to the possibility of vulnerabilities such as data breaches and cyber threats. Cryptology safeguards the authenticity and secrecy of delicate and classified data, shielding it from illegal intrusion. Ongoing research is being conducted to address emerging challenges in the field of computational difficulties and vulnerabilities in systems, which have arisen as a result of advancements and contributions to the existing body of knowledge. Due to advancements in technology and recent research, the introduction of quantum computing has revolutionized the field of science and prompted a reassessment of current cryptography methods.

E-mail address: ebru.yalcin305@gmail.com, fidan.nuriyeva@deu.edu.tr(*), erdem.alkim@deu.edu.tr.
Key words and phrases. Lattice-based cryptography, Polynomial multiplication algorithms, Number theoretic transform, Bruun algorithm.

Shor's algorithm is a quantum algorithm that provides a polynomial solution to the discrete logarithm problem, which is used in the current cryptographic protocol [1]. Shor's algorithm is a significant technique that utilizes quantum computing principles to perform operations on big integers, namely for factoring and solving fractional logarithms. These mathematically challenging problems seem to serve as the foundation for numerous encryption methods. Shor's algorithm is a prominent method that leverages the concepts of quantum computing to carry out computations on large integers, namely factoring and solving fractional logarithms. These mathematically complex difficulties appear to be the basis for many encryption systems. Shor's method presents a substantial risk to the security of widely utilized public key cryptosystems such as RSA and ECC. In 2018, the National Institute of Standards and Technology (NIST) in the United States launched a standardization project to tackle these emerging challenges. Developed specifically to provide a long-term defense against quantum computers, these innovative techniques are based on universally accepted mathematical problems that are difficult for both classical and quantum technology to solve [2].

Lattice-based systems are the most promising and prominent approach among recently developed systems. Lattice-based systems are notable due to the elevated complexity of lattice problems, which come from the challenging nature of mathematical issues. Their characteristics enable lattice-based systems to offer a resilient encryption mechanism and an effective defense against attacks. Lattice-based systems perform computations on polynomial rings. The primary benefit of utilizing these systems operating on polynomial rings lies in their inherent algebraic structure, which enables rapid expression of polynomial coefficients and proper execution of operations. The efficient storage of polynomial coefficients and the facilitation of effective operations are made feasible by this structure [3]. While lattice-based systems offer numerous advantages, polynomial multiplication is a computationally expensive operation. The computational load of processing the polynomials increases significantly due to the quick increase in multiplication complexity, which depends on the degree of the polynomials. Novel polynomial multiplication algorithms have been suggested to address this issue. These novel multiplication algorithms employ several techniques to decrease the computational complexity of the point-wise multiplication process and enhance and optimize overall efficiency.

The primary instances of these multiplication algorithms include the School-Book, Karatsuba, Toom-Cook, The Number Theoretic Transform (NTT), and Toeplitz Matrix-Vector Multiplication (TMVP) algorithms. The School-Book algorithm is the most fundamental and commonly used method for polynomial multiplication in literature. This algorithm is implemented by performing a straight multiplication of two polynomials. Karatsuba is a multiplication algorithm that reduces the total number of multiplications by employing the divide-and-conquer approach. It accomplishes this by dividing the polynomials into smaller segments while executing the multiplication. The NTT algorithm is a mathematical transformation method derived from the Fast Fourier Transform (FFT). It is an enhanced version of the FFT that has been further developed using number field theory. The NTT algorithm is mostly used for polynomial multiplication. This study also investigates the TMVP method, which is a specific algorithm that exploits the Toeplitz matrix structure commonly encountered in lattice-based systems. The Method section examines a polynomial multiplication algorithm known as the Bruun algorithm [4].

These multiplication algorithms are seen to be used on many different schemes today. For example, Kyber [5], Falcon [6], and Dilithium [7], among the projects that made it to NIST's standardization competition final in post-quantum cryptography are lattice-based systems. These schemes use polynomial multiplication extensively in their different stages and aim to ensure efficiency and security. They aim to speed up polynomial multiplications and reduce the complexity of the operation by using polynomial multiplication algorithms such as NTT and FFT. In this way, large-degree polynomials can be operated on, allowing complex calculations to be made. Additionally, these algorithms appear to produce accurate and reliable results. Due to these features, it appears to reduce the load on the processor and optimize energy consumption. For lower-power devices and embedded systems, these features are important. Thus, the schemes used in post-quantum cryptography are expected to work successfully in real-life applications.

Lattice-based cryptography has become a leading candidate for post-quantum security due to its robustness and reliance on complex mathematical problems. A critical aspect of these systems is polynomial multiplication, a resource-intensive operation that significantly impacts performance. Efficient algorithms such as NTT and TMVP play a vital role in optimizing cryptographic schemes like NTRU. This study focuses on improving polynomial multiplication to enhance the efficiency and practicality of post-quantum cryptographic systems for real-world applications.

In this study, information is given on polynomial multiplication algorithms used in lattice-based systems. In section two, firstly, the definition and mathematical representation of the polynomial ring and the definition of polynomial multiplication are given. In the same section, polynomial multiplication algorithms frequently used in lattice-based systems; NTT algorithm, and TMVP algorithm were examined. In the Third section, Bruun's algorithm is introduced. In chapter four, the results are given. Finally, in chapter five, we conclude our paper.

2. Polynomial Multiplication

Polynomial multiplication refers to the process of multiplying two polynomials inside the same polynomial ring. The current scenario can be expressed in the following manner.

Definition 2.1. Let $\mathscr{R}$ be an accumulative ring, $N \in \mathbb{N}$, and $0 \leq i < N$. Let $a_i, d_i, c \in \mathscr{R}$ be coefficients. The polynomials $a(x)$ and $d(x)$ are subjected to the polynomial multiplication operation within the same polynomial ring, resulting in:

$$c = a(x) \cdot d(x) \tag{1}$$

$$c_k = \sum_{i=0}^{k} a_i d_{k-i} + \sum_{i=k+1}^{N-1} a_i d_{N+k-i} = \sum_{\substack{j+i \equiv k \\ (\text{mod } N)}} a_i d_j \tag{2}$$

The polynomial multiplication operation takes place in the ring $\mathscr{R} = \mathbb{Z}_q/(x^N - 1)$, and the factors and product elements become elements of the ring $\mathscr{R} = \mathbb{Z}_q/(x^N - 1)$. Specifically, if $2x+1 = c_1$, then $2x+1$ serves as a factor and a product element within the ring $\mathscr{R} = \mathbb{Z}_q/(x^N - 1)$ [8].

Polynomial multiplication is typically carried out on polynomials with high degrees. Multiplication, which is one of the arithmetic operations carried out on polynomials, requires a greater amount of time and computational power compared to other operations. Consequently, researchers have conducted investigations to enhance this circumstance by developing polynomial multiplication algorithms. These algorithms are implemented on various systems based on specific requirements. Polynomial multiplication algorithms can be categorized as follows: School-Book, Toom-Cook, Number Theoretic Transform, Toeplitz Matrix-Vector Product, and Bruun.

The subsequent part delves into a thorough examination of polynomial multiplication algorithms, analyzing each one individually. Explicit formulations and efficient algorithms are provided.

2.1. **Number Theoretic Transform:**

The Number Theoretic Transform (NTT) is a mathematical technique mostly employed for solving the factorization issue. Its output is derived from the Fast Fourier Transform. According to [9] it is also claimed that this is a version of DFT that operates on finite fields rather than complex numbers. The method originated as an extension of the FFT, although its precise output remains uncertain. Several mathematicians and computer scientists made significant contributions to the initial investigations of NTT. S.S. Winograd introduced a polynomial evaluation procedure utilizing the Chinese remainder theorem [10] after examining the literature. The investigation led to the invention of NTT, which focuses on the remaining parts based on prime numbers. Using the findings from conducted studies, an algorithm was designed that utilizes CRT and modular arithmetic operations to accomplish polynomial multiplication, incorporating novel features. He enhanced the development of NTT's applications by incorporating this algorithm. NTT has achieved its current level of usage through the contributions of fundamental research areas like FFT and modular arithmetic. These areas have been crucial in developing efficient algorithms for polynomial multiplication and polynomial factorization.

NTT is a mathematical operation called the fractional Fourier transformation, which is defined on the ring $\mathscr{R}_q = \mathbb{Z}_q/\Phi_m(x)$. It performs fast calculations on polynomials, hence improving the efficiency of polynomial multiplication. A polynomial $a(x)$ over the ring $\mathscr{R}_q$, with a degree of $n-1$, can be represented as:

$$a(x) = \sum_{i=0}^{n-1} a_i x^i \tag{3}$$

The NTT of a polynomial $\bar{a}(x)$ of degree $n-1$ on the ring $\mathscr{R}_q$ is represented in polynomial form as:

$$\bar{a} = \sum_{i=0}^{n-1} \bar{a}_i x^i \tag{4}$$

where the coefficients $\bar{a}$ can be defined using the following (2.38):

$$\bar{a} = \sum_{j=0}^{n-1} \bar{a}_j w^{i \cdot j} \pmod{q} \quad \text{for } i = 0, 1, 2, \ldots, n-1. \tag{5}$$

The equation involves a twiddle factor, denoted as ω, which must satisfy the criteria $\omega^n \equiv 1 \pmod{q}$, and $\omega^i \neq 1 \pmod{q}$ for all $i < n$. The NTT operation is executed by computing this equation for every value of i ranging from 0 to $n-1$.

The NTT operation employs a constant known as the twiddle factor, $\omega \in \mathbb{Z}_q$, which represents the n-th root of unity. The method utilizes a basic n-th root of unity, $\omega \in \mathbb{Z}_q$, which fulfills the requirements $\omega^n \equiv 1 \pmod{q}$, $\omega^i \not\equiv 1 \pmod{q}$ for all $i < n$, and $q \equiv 1 \pmod{n}$. The inverse Number Theoretic Transform (INTT) operation follows a similar method, except in the last step, the element $\omega^{-1} \in \mathbb{Z}_q$ is employed instead of ω. Furthermore, in the mathematical field $\mathbb{Z}_q$, while performing the final step of the Inverse Number Theoretic Transform (INTT) calculation, the resulting coefficients are multiplied by the inverse of n^{-1} [11].

2.2. **Toeplitz Matrix-Vector Product:**

The TMVP algorithm is a highly efficient method for multiplying matrices and vectors, specifically designed to exploit the distinctive characteristics of Toeplitz matrices. The actual origin and date of the proposal for TMVP are uncertain, although it is believed that the concept of utilizing Topelitz matrices for efficient computations is derived from the research conducted by Otto Toeplitz. German mathematicians specialized in the areas of algebraic and numerical analysis. Toeplitz matrices are square matrices whose each diagonal has constant values. These matrices possess mathematical properties that make them useful for efficient calculations [12]. Subsequent works further investigated and elaborated on the concept of utilizing Toeplitz matrices for polynomial multiplication. Utilizing Toeplitz matrices aids in diminishing the overall intricacy through the pre-calculation and reuse of certain pieces. Furthermore, when multiple multiplication operations are required on a single matrix, TMVP executes these operations efficiently by avoiding redundant calculations, hence greatly enhancing efficiency. Given this circumstance, utilizing it in polynomial multiplication operations, which are crucial in lattice-based systems prevalent in post-quantum cryptography, offers several benefits. Toeplitz matrices are commonly employed in various cryptographic applications, as evidenced by their frequent appearance in the literature [13], [14], [15].

A TMVP (n-dimensional) can be computed by utilizing three TMVPs ($n/2$-dimensional) in a 2-way TMVP formula. Denote the half-dimensional divisions of T as T_0, T_1, and T_2, and the half-dimensional partitions of the vector V as V_0 and V_1. The calculation of the N-dimensional matrix-vector multiplication is performed in the following manner:

$$T \cdot V = \begin{pmatrix} T_1 & T_0 \\ T_2 & T_1 \end{pmatrix} \begin{pmatrix} V_0 \\ V_2 \end{pmatrix} = \begin{pmatrix} P_0 + P_1 \\ P_0 - P_2 \end{pmatrix} \tag{6}$$

and

$$
\begin{aligned}
P_0 &= T_1(V_0 + V_1), \\
P_1 &= (T_0 - T_1)V_1, \\
P_2 &= (T_1 - T_2)V_0
\end{aligned} \tag{7}
$$

The complexity of TMVP_2 can be expressed as

$$M_{\text{TMVP}_2}(n) = 3M(n/2) + 3n - 1$$

based on the operations mentioned above. Furthermore, an n-dimensional Total Mean Value Projection (TMVP) can be computed by utilizing three $n/3$-dimensional TMVPs in a 3-way TMVP equation. By following the same procedure as for TMVP_2, we can determine the existence of TMVP_3. The complexity of TMVP_3 can be expressed as

$$M_{\text{TMVP}_3}(n) = 6M(n/3) + 5n - 1.$$

It is evident that when performing polynomial multiplication, various formulas are generated for TMVPs based on their size and effectiveness. The article [16] provides a more detailed analysis of the process of developing additional TMVP formulas using the given formulas mentioned above.

2.2.1. *Polynomial Multiplication Modulo $x^n \pm 1$ via TMVP.*

In the polynomial multiplication operation, when performed on $\mathbb{Z}[x]/(x^n \pm 1)$, the resulting polynomial $\mathbb{Z}[x]$ should be reduced by using $x^n \pm 1$. TMVP leverages the structure of Toeplitz matrices and the properties of polynomial multiplication to optimize the operations to be executed. TMVP utilizes polynomial multiplication operations on $x^n \pm 1$ to answer a wide range of issues across several disciplines. The modulo operation is of great importance in the coding and decoding procedures of Error-correcting codes as it enables the identification and correction of errors. Cryptography employs it in several systems, including digital signatures and encryption algorithms. It offers a highly effective computational capability for the systems in which it is employed. Since reduction modulo $x^n \pm 1$ is only a addition or subtraction T_2 becomes $\pm T_0$, thus equ. 6 and equ. 7 becomes:

$$T \cdot V = \begin{pmatrix} T_1 & T_0 \\ \pm T_0 & T_1 \end{pmatrix} \begin{pmatrix} V_0 \\ V_2 \end{pmatrix} = \begin{pmatrix} P_0 + P_1 \\ P_0 - P_2 \end{pmatrix} \tag{8}$$

and

$$
\begin{aligned}
P_0 &= T_1(V_0 + V_1) \\
P_1 &= (T_0 - T_1)V_1 \\
P_2 &= (T_1 \pm T_0)V_0
\end{aligned} \tag{9}
$$

3. Factorization of the Cyclotomic polynomial $x^{2^k}+1$

The Bruun technique, as described by George Bruun in his 1978 publication, is a Discrete Fourier Transform algorithm specifically designed for real numbers with a logarithmic basis [17]. These processes, which are associated with the traditional complex FFT, allow for the utilization of new FFT variations that exclusively operate with real coefficients. Additionally, the implementation of new FFT algorithms involves utilizing only half the number of real multiplications compared to existing FFT methods.

The Bruun method is the method employed to ascertain the factors of a polynomial that encompasses all unit roots. This method utilizes the structural characteristics of the unit roots of polynomials to expedite the computation of polynomial factors. This approach, employed in areas such as number theory and modular arithmetic, is said to enhance the efficiency of polynomial calculations in post-quantum cryptography. Bruun's approach is designed to factorize the roots of a polynomial of degree n. It achieves this by recursively finding the explicit roots of the polynomial.

The new structure demonstrates a logarithmic reduction in calculation time and achieves processing efficiency by dividing the DFT operations into segments and executing certain parallel operations. Therefore, it can be stated that intricate discrete Fourier transform (DFT) processes have experienced an increase in efficiency. In the classical approach of FFT, $N/2\log N$ complex multiplication operations are performed, where two complex numbers are multiplied in each operation. The new method, in contrast to the old one, was demonstrated to involve the multiplication of a real number and a complex number.

Using Bruun's algorithm, the same outcome as the classical technique can be achieved by employing the multiplication of real and complex numbers instead of complex multiplications. It is evident that the new algorithm has decreased the utilization of intricate multiplication in the classical way by half. Therefore, it is widely acknowledged that the Bruun algorithm operates with greater speed and efficiency.

This section demonstrates the complete separation of the polynomial $x^{2^k}+1$ on $\mathbb{F}_p$ into separable polynomials. p is a prime number that fulfills the criterion $p \equiv 3 \pmod 4$. Therefore, it is demonstrated that it is possible to create an irreducible polynomial over $\mathbb{F}_p$ with a degree that is a power of 2. Therefore, it is evident that this approach can be effectively utilized in FFT applications within limited regions.

The following theorem pertains to the situation when p is a prime integer and is entirely irreducible on $\mathbb{F}_p$, subject to the constraints of $p \equiv 3 \pmod 4$, meaning $p \equiv -1 \pmod{2^{k+1}}$ [18].

Let p be a prime number that satisfies $p \equiv 3 \pmod 4$. The process of complete factorization of the polynomial $x^{2^k}+1$ on the $\mathbb{F}_p$ field is investigated. To simplify the problem, it can be seen that the roots of the polynomial $x^{2^k}+1$ are actually the primitive $x^{2^k}+1$-th roots of unity in an expansion field of $\mathbb{F}_p$. Therefore, the goal is to construct the smallest polynomials on $\mathbb{F}_p$ for primitive $x^{2^k}+1$-th roots of unity where k is an integer greater than or equal to 1. If we consider that the degree of every i

Let the highest exponent of 2 in $p+1$ be denoted as 2^a. The expression p^2-1, where p is a variable, represents the maximum power of 2 and is denoted as 2^{a+1}. Assuming $\alpha \in \mathbb{F}_{p^2}$, let's consider that it has a degree of 2^{a+1}. It should be noted that the polynomial $x^{2^e}+\alpha$ is irreducible over the field $\mathbb{F}_{p^2}$ for $e \geq 0$. According to this information, the 2^{e+1}-th order cannot be broken down on $\mathbb{F}_p$, and the primitive roots

of the 2^{a+e+1}-st order are given as $(x^{2^e}+\alpha)(x^{2^e}+\alpha^p)$. The article states that $\mathbb{F}_{p^2}=\mathbb{F}_p(i)$, where i is the square root of -1. Additionally, since f is defined as $\mathbb{F}_{p^2}\to\mathbb{F}_{p^2}$, it is expressed as $(1+x)^{\frac{p-1}{2}}$ [19], [20].

The following formula provides the steps for calculating the square root in $\mathbb{F}_{p^2}$. The formula applicable to any second-order α residue in $\mathbb{F}_p(i)$ is given as follows.

$$\sqrt{\alpha}=\begin{cases} i\alpha^{\frac{p+1}{4}}, & \text{if } \alpha^{\frac{p-1}{2}}=-1,\\ \left(1+\alpha^{\frac{p-1}{2}}\right)^{\frac{p+1}{2}}\alpha^{\frac{p+1}{4}}, & \text{otherwise.}\end{cases} \tag{10}$$

If k is greater than 0, and the order of the element α is 2^k, both $\sqrt{\alpha}$ and $\sqrt{-\alpha}$ have an order of $x^{2^k}+1$. If we define the starting point as $i=\sqrt{-1}$, it becomes evident that the numbers with an exponent of 2^k can be determined using a recursive process. Prior to factoring the polynomial $x^{2^k}+1$, it is essential to calculate the minimum polynomials of all the elements produced in $\mathbb{F}_p$.

Theorem 3.1. *Let $H_1=\{0\}$.*

$$H_k=\pm\left\{\left(\frac{u+1}{2}\right)^{\frac{p+1}{4}}\right\}\quad \textit{for } u\in H_{k-1} \tag{11}$$

For every value of k from 1 to $a-1$, the cardinality of H_k is equal to 2^{k-1},

$$x^{2^k}+1=\prod_{u\in H_k}\left(x^2-2ux+1\right) \tag{12}$$

For any integer $e\geq 0$,

$$x^{2^k}+1=\prod_{u\in H_k}\left(x^{2^{e+1}}-2ux^{2^e}-1\right) \tag{13}$$

The aforementioned theorem can be used for additional cyclotomic polynomials; however, it should be noted that these polynomials are not directly connected to the Bruun paper. Nevertheless, there exists a connection between them. Bruun's Algorithm utilizes polynomial factorization to conduct DFT computations. This is analogous to the factorization of cyclotomic polynomials, as both approaches involve dividing polynomials into smaller components. The relationship between Bruun's algorithm and cyclotomic polynomials is established by the factorization of polynomials and the utilization of unit roots. This connection offers enhanced efficiency and rapidity in DFT calculations and polynomial factorizations.

4. Results

The concluding part is organized into three distinct topics. Initially, an analysis was conducted on the number of multiplications and their complexity, which are determined by specific parameters (n), for the multiplication operation. This operation is known to be the most time-consuming and costly among the various operations performed on polynomial rings. The second title provides a general explanation of the Bruun approach and includes some inferred information. In the last heading provides details on

polynomial multiplication methods and discusses their efficiency.

4.1. **Multiplication Algorithms:**

Below is a table quoted from the article [21], examining the number of multiplications and time complexities in multiplication algorithms depending on certain n parameters. In this table, one of the frequently used polynomial multiplication algorithms; Scool-Book, Karatsuba, Toom-Cook-way, TMVP_2, TMVP_3, TMVP_4 and NTT algorithms were examined.

TABLE 1. **Complexities of multiplication algorithms**

No	Multiplication Algorithms	Complexity
1	School-Book	$T(n) = 2n^2 - 2$
2	Karatsuba	$T(n) = 3T(n/2)$
3	Toom-Cook-k	$T(n) = (2k-1)T(n/3)$
4	TMVP-2	$T(n) = 3T(n/2) + 3n - 1$
5	TMVP-3	$T(n) = 6T(n/3)$
6	TMVP-4	$T(n) = 7T(n/4) + 5n - 1$
7	NTT	$T(n) = \frac{3}{2}n\log n + n$
8	Bruun	$T(n) = \frac{3}{2}n\log n + n$

Using the information in Table 1, some inferences for multiplication algorithms for certain parameters are given in Table 2 below.

We have chosen a specific parameter set of NTRU, known as `ntruhrss701`, for the purpose of comparing algorithms. In this parameter set, the value of q is 2^{13}, n is 701, and $f(x)$ is defined as $x^{701} - 1$. Consequently, the polynomial ring can be represented as $\mathbb{Z}_{2^{13}}[x]/(x^{701} - 1)$.

To carry out multiplication in the ring $\mathbb{Z}_{2^{13}}[x]/(x^{701} - 1)$, TMVP must divide the input polynomials. Given that n is a prime integer, it is necessary to select a size that is larger than n to perform the multiplication calculation. Since the majority of implementations focus on sizes in the form of $a' = 2^k 3^l t$, where t is less than 16 for optimal performance, we computed the number of recursion steps for the two shortest possibilities within the chosen polynomial ring.

$$704 \xrightarrow{\text{TMVP}_4} 176 \xrightarrow{\text{TMVP}_2} 88 \xrightarrow{\text{TMVP}_2} 44 \xrightarrow{\text{TMVP}_2} 22 \xrightarrow{\text{TMVP}_2} 11 \tag{14}$$

$$720 \xrightarrow{\text{TMVP}_4} 180 \xrightarrow{\text{TMVP}_3} 60 \xrightarrow{\text{TMVP}_3} 20 \xrightarrow{\text{TMVP}_2} 10 \tag{15}$$

The $T(n')$ values in Table 2 were computed based on the complexity provided in Table 1. The number of cycles reported by [15], [22], and [23] were all measured on the ARM Cortex-M4 Discovery board, which served as the common target platform. It can be seen that Table 2 below was created using the parameters 704, 720, 1440, and 1536.

Table 2 comprises $T(n)$ values computed based on the time complexity algorithms provided in Table 1. Upon examination and comparison of the TMVP, NTT, and Toom-Cook multiplication algorithms, it is evident that NTT-based polynomial multiplication necessitates fewer operations, even for dimensions of $2x$. However, it is important to note that NTT does require modular arithmetic operations. In contrast, the TMVP and Toom-Cook algorithms are capable of operating with two basis powers, eliminating the need for additional modular reduction following elementary arithmetic operations. While the TMVP algorithm incorporates polynomial reduction in its multiplication operations, it necessitates fewer operations.

TABLE 2. **Multiplication Cycles for TMVP, NTT, and Toom-Cook Algorithms**

	TMVP		NTT		Toom-Cook	
n'	$T(n')$	#Cycles	$T(n')$	#Cycles	$T(n')$	#Cycles
704	115331	142252 [24]	-	-	68607	172882 [22]
720	125519	-	-	-	52500	-
1440	-	-	45360	141000 [23]	-	-
1536	-	-	42240	148000 [23]	-	-

This study examines various multiplication algorithms utilized to enhance the efficiency of Lattice-based cryptographic protocols. It provides comparisons of these algorithms based on their theoretical complexity and current usage in the field. An assessment was conducted based on certain criteria on the NTRU scheme, which is one of the lattice-based systems that reached the final stage in the standardization competition hosted by NIST [21].

Lattice-based systems have excelled among the various schemes in the competition set by NIST in the field of post-quantum cryptography. These systems operate on polynomial rings. To optimize the efficiency of the systems, several novel polynomial multiplication algorithms have been developed to minimize the number of multiplication operations required. The algorithms used in this study are Karatsuba, Toom-Cook, NTT, TMVP, and the Bruun polynomial multiplication algorithm mentioned in the technique section.

When the research on polynomial multiplication algorithms in the literature is generally examined, it is seen that various comparisons and evaluations have been made for these algorithms in terms of efficiency, security, and suitability for different applications. Lattice-based algorithms such as Karatsuba, Toom-Cook, NTT, and TMVP will be used in post-quantum cryptography. It can be said that it is among the polynomial multiplication algorithms that are thought to be very important for systems. It is obvious that the proposed algorithms will bring new features, optimization techniques, and efficiency improvements to this field.

5. CONCLUSION

In this study, multiplication operations performed in lattice-based systems on polynomial rings in post-quantum cryptography are examined. It is obvious that the operation that causes the most time

and cost among the operations on polynomial rings is the multiplication operation. If the value of the parameter n is chosen as 701 for the NTRU scheme, it has been shown that the NTT-based polynomial multiplication algorithm exhibits the highest performance, despite its higher memory usage. TMVP-based polynomial multiplication algorithms have demonstrated superior memory efficiency compared to other algorithms, resulting in only a 1% decrease in multiplication operations. Bruun's algorithm, another suggested approach, achieves the same outcome as other methods by performing multiplication operations on complex and real values. As a future work, we plan to compare the implementation of the Bruun algorithm with NTT for the same polynomial ring.

DECLARATIONS

- **Contribution Rate Statement:** Ebru Yalcin: writing – original draft, resources, methodology, conceptualization, Fidan Nuriyeva: writing – review & editing, methodology, Erdem Alkım: writing – review & editing, methodology.
- **Conflict of Interest:** The authors have not disclosed any competing interests.
- **Data Availability:** Data sharing is not applicable to this article as no datasets were generated or analyzed.
- **Statement of Support and Acknowledgment:** None.

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.

[2] G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, C. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, D. Smith-Tone, Y.-K. Liu, Status report on the first round of the NIST post-quantum cryptography standardization process, National Institute for Standards and Technology Internal Report 8240, `https://doi.org/10.6028/NIST.IR.8240` (2019).

[3] D. Micciancio, O. Regev, Lattice-based cryptography, in: Post-quantum cryptography, Springer Berlin Heidelberg, 2009, pp. 147–191.

[4] V. Hwang, A survey of polynomial multiplications for lattice-based cryptosystems, Cryptology ePrint Archive, Paper 2023/1962 (2023).
URL `https://eprint.iacr.org/2023/1962`

[5] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, D. Stehlé, Crystals-kyber: a cca-secure module-lattice-based kem, in: 2018 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2018, pp. 353–367.

[6] L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé, Crystals–dilithium: Digital signatures from module lattices, Cryptology ePrint Archive (2018).

[7] P. A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, Z. Zhang, Falcon: Fast-fourier lattice-based compact signatures over NTRU, Submission to the NIST's post-quantum cryptography standardization process 36 (5) (2018) 1–75.

[8] R. T. Moenck, Practical fast polynomial multiplication, in: Proceedings of the third ACM symposium on Symbolic and algebraic computation, ACM, 1976, pp. 136–148.

[9] D. Harvey, Faster arithmetic for number-theoretic transforms, Journal of Symbolic Computation 60 (2014) 113–119.

[10] S. Winograd, On computing the discrete fourier transform, Mathematics of computation 32 (141) (1978) 175–199.

[11] K. Derya, A. C. Mert, E. Öztürk, E. Savaş, CoHA-NTT: A configurable hardware accelerator for NTT-based polynomial multiplication, Microprocessors and Microsystems 89 (2022).

[12] O. Toeplitz, Das algebraische analogon zu einem satze von fejér, Mathematische Zeitschrift 2 (1) (1918) 187–197.
[13] S. Ali, M. Cenk, Faster residue multiplication modulo 521-bit mersenne prime and an application to ECC, IEEE Transactions on Circuits and Systems I: Regular Papers 65 (8) (2018) 2477–2490.
[14] M. A. Hasan, N. Meloni, A. H. Namin, C. Negre, Block recombination approach for subquadratic space complexity binary field multiplication based on toeplitz matrix-vector product, IEEE Transactions on Computers 61 (2) (2010) 151–163.
[15] I. K. Paksoy, M. Cenk, TMVP-based multiplication for polynomial quotient rings and application to saber on ARM cortex-M4, cryptology ePrint Archive (2020).
[16] S. Winograd, On multiplication of polynomials modulo a polynomial, SIAM Journal on Computing 9 (2) (1980) 225–229.
[17] G. Bruun, z-transform DFT filters and FFT's, IEEE Transactions on Acoustics, Speech, and Signal Processing 26 (1) (1978) 56–63.
[18] I. F. Blake, S. Gao, R. C. Mullin, Explicit factorization of $x^{2^k}+1$ over F_p with prime $p \equiv 3 mod 4$, Applicable Algebra in Engineering, Communication and Computing 4 (2) (1993) 89–94.
[19] H. W. Lenstra, Finding isomorphisms between finite fields, Mathematics of Computation 56 (193) (1991) 329–347.
[20] V. Shoup, New algorithms for finding irreducible polynomials over finite fields, Mathematics of computation 54 (189) (1990) 435–447.
[21] E. Yalçın, F. Nuriyeva, E. Alkım, A comparative study on polynomial multiplication algorithms in context on post-quantum cryptography, in: DEU International Symposium Series on Graduate Researches-2022 DataScience, DEU, 2022, pp. 1–10.
[22] M. Kannwischer, P. Bissmeyer, S. Schmidt, Optimizing lattice-based cryptography schemes with structured noise, in: Post-Quantum Cryptography: 5th International Conference, PQCrypto 2019, Fukuoka, Japan, July, Springer, 2019, pp. 81–97.
[23] E. Alkim, V. Hwang, B.-Y. Yang, Multi-parameter support with ntts for ntru and ntru prime on cortex-m4, IACR Transactions on Cryptographic Hardware and Embedded Systems 2022 (4) (2022) 349–371.
[24] I. K. Paksoy, M. Cenk, Faster ntru on arm cortex-m4 with tmvp-based multiplication, IEEE Transactions on Circuits and Systems I: Regular Papers 69 (10) (2022) 4083–4092.