NTT quantum circuit for CRYSTALS–Kyber

Gyeongju Song*
Hansung University
Seoul, South Korea
thdrudwn98@gmail.com

Kyungbae Jang*
Hansung University
Seoul, South Korea
starj1023@gmail.com

Siwoo Eum*
Hansung University
Seoul, South Korea
shuraatum@gmail.com

Minjoo Sim*
Hansung University
Seoul, South Korea
minjoos9797@gmail.com

Hwajeong Seo†
Hansung University
Seoul, South Korea
hwajeong84@gmail.com

Abstract

The advent of quantum computers is a threat to the existing cryptosystem, and NIST is preparing for the post-quantum era through the PQC contest. Crystal-Kyber is a lattice-based cipher suite that advanced to the 4 round of the PQC standardization process conducted by NIST. Lattice-based cryptography is considered safe for quantum computer because a quantum algorithm that can solve the lattice problem of lattice-based cryptography more efficiently than classic algorithm is not yet known. In this paper, we propose a quantum circuit for NTT used for efficient polynomial multiplication of CRYSTALS-Kyber. The proposed CRYSTALS-Kyber NTT quantum circuit operates at $Z_{3329}[X]/(X^{256}+1)$. We describe the operation of the NTT quantum circuit in detail and finally estimate the quantum resources required for the NTT operation. As far as we know, this is the first implementation of the CRYSTALS-Kyber NTT quantum circuit. It is expected that our attempts will contribute to analyzing the security strength of quantum computers for lattice-based cryptography.

1 Introduction

The advent of large-scale quantum computers is expected to pose a threat to the current cryptosystem. Public key cryptography maintains security based on the difficulty of factoring large integers. However, Shor's quantum algorithm is known to pose a threat to public key cryptography because it is possible to factor large integers within polynomial time [11]. Symmetric key cryptography maintains the security strength depending on the key length. Grover's algorithm operate a $\sqrt{2^n}$ query for a symmetric key cryptography with an n-bit key to accelerate a brute force attack [6]. In recent years, research have been progressed to verify the cryptographic strength through the estimation of quantum resources required for quantum algorithms [9, 1, 13, 12, 8, 2, 7]. When the available resources of a quantum computer reach the resources required for a cryptographic attack, it is expected that the target Cryptography will be broken. Therefore, it is expected that existing cryptosystems will be replaced by Post-Quantum Cryptography (PQC) algorithms. The National Institute of Standards and Technology (NIST) conducted standardize PQC algorithms to replace the existing cryptosystem in preparation for the post-quantum era [4].

In this paper, we propose a quantum circuit for number theoretic transform (NTT) used in CRYSTALS-Kyber. CRYSTAL-Kyber [3] is an IND-CCA2-secure KEM with the hardness of Module-LWE [10] on a lattices. In sub operation of CRYSTALS-Kyebr, polynomial multiplication operation is performed with n=256 and q=3329 parameters for $Z_q[X]/(X^n+1)$. About this, NTT is used to efficiently perform

G. Sutcliffe, A. Voronkov (eds.): easychair 1.0, 2008, volume 1, issue: 1, pp. 1-8

*Did all the difficult work

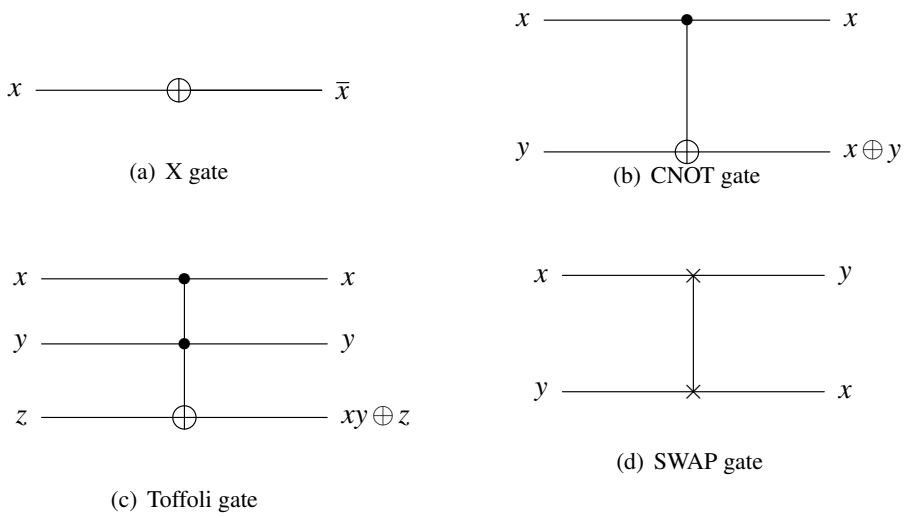
†Corresponding Author

modular N multiplication for polynomials a and b. General school-book multiplication has a computational complexity O(n^2), but multiplication through NTT transformation has a computational complexity O($nlogn$). As far as we know, the proposed quantum circuit is the first NTT quantum circuit designed to operate at the parameters of CRYSTALS-Kyber. Our quantum circuit implemented NTT multiplication for $Z_q[X]/(X^n+1)$ to operate at CRYSTALS-Kyber parameter n =256, q =3329. We implemented a negative integer by expressing it in two's complement in qubit, and $n \times 32$ qubits are used to express $Z_q[X]/(X^n+1)$ coefficients. Since we express the intermediate computation process in two's complement, we don't allocate extra qubits to determine negative and positive integers. However, only 1-bit qubits are used for the intermediate conditionals. Finally, we estimate and analyze the quantum resources required for proposed CRYSTALS-KYBER NTT

2 Background

2.1 Quantum computing

Quantum computing should be reversible for all changes(noise, computation, etc.) except measurements. Reversible means that the initial state can be regenerated from the output state without additional information. In order to have such a reversible characteristic, the number of input/output bits at the gate should be the same. Figure 1 shows quantum gates with reversible property. Quantum computing should be reversible for all changes(noise, computation, etc.) except measurements. Reversible means that the initial state can be regenerated from the output state without additional information. In order to have such a reversible characteristic, the number of input/output bits at the gate should be the same. Figure 1 shows quantum gates with reversible property.



(a) X gate (b) CNOT gate

(c) Toffoli gate (d) SWAP gate

Figure 1: Quantum gates

(a) **X gate**: The X gate operates with a single qubit and reversed the input state.

(b) **CNOT gate**: The CNOT gate operates with two qubits, which are divided into control qubit and target qubit. The state of the target qubit y is reversed only when the control bit x is one.

(c) **Toffoli gate**: The Toffoli gate operates with three qubits, divided into two control qubits and one destination qubit. The state of the target qubit z is reversed only when the control bits x and y are both one.

(d) **SWAP gate**: SWAP gate change the physical location of two qubits.

2.2 CRYSTALS-Kyber

CRYSTAL-Kyber [3] is an IND-CCA2-secure KEM with the hardness of Module-LWE [10] on a lattices. The Kyber cipher, designed to be robust in the post-quantum era, is one of the finalists of the post-quantum cryptography project conducted by NIST. Security is based on the hardness of resolving learning-with-errors (LWE) problems for module lattices.

2.3 Number theoretic transform(NTT)

Fourier transform (FT) is a mathematical transformation that decomposes a function of space or time into a function of space or time frequency. When there is an oscillation function $f(t)$ that varies with time t, the oscillation contains several frequency components. If the frequency component is extracted and expressed as the frequency intensity distribution $F(w)$ for the frequency w, the nature of vibration can be easily analyzed. The mathematical expression for FT follows:

$$F(\omega) = \frac{1}{\sqrt{2\pi}} = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

Discrete fourier transform (DFT) performs transformation on finite N complex number fields instead of continuous interval $(-\infty,\infty)$ of FT. The N complex number sequence $x_N = x_0, ..., x_{(n-1)}$ is converted into another complex number sequence $X_k = X_0, ..., X_t$. Equation 1 shows the DFT process of a complex number field.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n/N} \tag{1}$$

The Number Theory Transform (NTT) is a generalization of the discrete Fourier transform (DFT) domain to integer fields. It uses the n-th primitive root of unity based on a quotient ring instead of the complex field of DFT. Here, $j \equiv x^n \bmod q$ (x:generator of the multiplicative group, q:prime number, b:integer) is used instead of the complex number field $j = e^{\frac{-2\pi i}{n}}$. When performing multiplication on two n-bit length polynomials, typical school-book multiplication has a computational complexity of O(n^2) whereas that of NTT multiplication is O($nlogn$). In Lattice-based ciphers, NTT is used for efficient multiplication.

3 Proposed method

In this paper, we propose a quantum circuit for NTT multiplication used in a lattice-based algorithm, which is a candidate algorithm for PQC. The proposed NTT quantum circuit is designed to operate with CRYSTALS–Kyber parameters $N = 256$ and $Q = 3329$ for multiplication on $Z_q[X]/(X^n+1)$. We operate by defining the NTT polynomial ring as $Z_{3329}[X]/(X^{256}+1)$. In the quantum circuit, an optimization technique is applied to reduce the quantum resources. The operation sequence of the proposed quantum circuit is as follows:

$$\text{NTT quantum circuit} = NTT\ sub \circ Montgomery\ reduce \circ fmul$$

- **fmul** : It multiplies the NTT input and the *zetas* value. In detail, it is divided into $fmul_1$ and $fmul_2$.
- **Montgomery reduce** : It performs montgomery reduce multiplication on the input$\times zeta$.

- **NTT sub** : It performs addition and subtraction for Montgomery reduction result and input.

We use two's complement to represent negative numbers in qubits. Since zetas is a pre-calculated value, there is no separate calculation. The NTT quantum circuit is performed using montgomery reduction. We allocate 32 qubits for each term to represent the multiplication result as a binary number. In the CRYSTALS-Kyber parameter n = 256, q = 3329, $32 \times n$-qubits are used to store the coefficients in the $Z_q[X]/(X^n+1)$. The original input must be used in the last NTT sub function, so the function proceeds while maintaining the input. Therefore, each function uses the temp qubit to store the operation result and proceeds to the next function. A description of each function follows:

3.1 fmul

The inner operation of the $fmul$ function is to multiply input and $zeta$. Since $zetas$ is a fixed constant, the number of qubits is reduced by performing input addition equal to the $zetas$ size without assigning a value to the qubit. In this quantum circuit, the ripple carry adder [5] proposed by Cuccaro et al. was used. The $fmul$ function is different in the way it operates in the first NTT cycle and other cycles(C). In the first cycle ($C = 1$), the sign of the input and $zeta$ is known, so the process of determining the sign is not necessary. On the other hand, in $C \geq 2$, the sign of the input needs to be checked, so the sign is determined using *check* of 1-qubit. On the other hand, in $C \geq 2$, since the sign of the input needs to be confirmed, 1-qubit *check* is used to confirmed the sign.

Algorithm 1 in Appendix shows the operation of the $fmul_1$ function when $C = 1$, and Algorithm 2 in Appendix shows the operation of the $fmul_2$ function when $C \geq 2$. Since the input must retain its original value, the function result is stored in 32-qubit $temp$. Both $fmul$ functions use CNOT gates to store input values in $temp$ and perform multiplication. In the $fmul$ function, the sign of the input and $zeta$ is checked. If the sign is the same, the result is positive, and if the sign is different, the result is negative. As a result, the value of (input$\times zetas$) is stored in $temp$ qubit.

3.2 Montgomery reduce

This function performs montgomery reduction multiplication on the input$\times zeta$. For each term, it is calculated on the $Z_q[X]/(X^n+1)$ field. The parameters of Kyber NTT are fixed as q=3329, n=256.

Algorithm 3 in Appendix shows the operation of the Montgomery reduce quantum circuit. In the for loop, Q is $q = 3329$ and QIV is the inverse of $Q : -3327$. Since Q and $QINV$ are known values, the result of multiplying by the corresponding size is obtained without allocating qubits to store the values of Q and $QINV$. Input $\times$ Q and Input$\times$ $QINV$ are quantum to classic operation, not quantum to quantum operation. Finally, index values [0] to [15] are discarded through 16-bit left shift and the values of indexes [16] to [31] are returned.

3.3 NTT sub

The NTT sub function performs addition and subtraction between the montgomery reduce result and the input having the corresponding index, and operates as NTT_{sub1} and NTT_{sub2} in detail. NTT_{sub1} and NTT_{sub2} compute the following: NTT_{sub1} = (input $-$ (Montgomery result)), NTT_{sub2} = (input + (Montgomery result)). In order to sequentially calculate the formula, both the original input and Montgomery result must be maintained after NTT_{sub1}. Since it is not possible to keep all of the calculation targets (input, Montgomery reduce result), the input is stored in temp qubits and the calculation is performed. Figure 2 shows the operation of the quantum circuit of NTT_{sub1} and NTT_{sub2}.

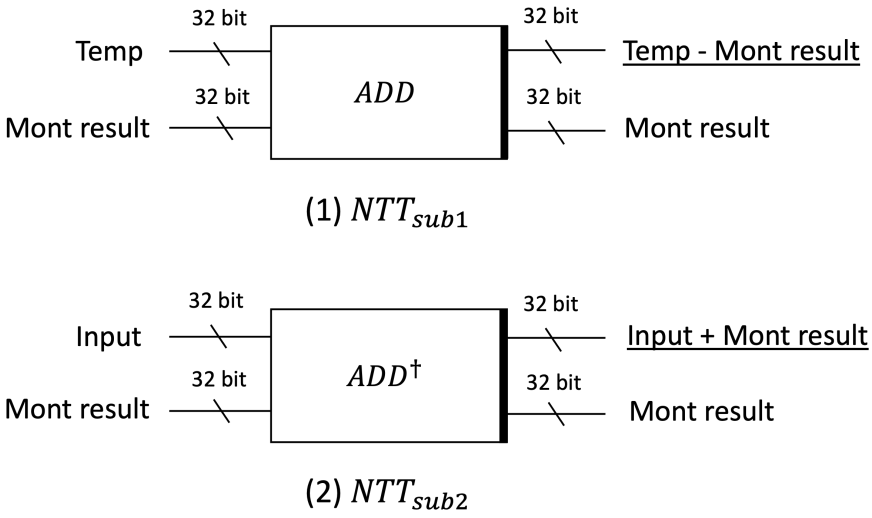


Figure 2: NTT sub operation(Above:(1)NTT_{sub1}, Below:(2)NTT_{sub2})

NTT_{sub1} stores the subtraction of temp and Mont result in temp, and Mont result is maintained. NTT_{sub2} stores the addition of input and Mont result in input. The results of all operations are sorted according to the NTT array index order.

4 Evaluation

We estimated quantum resources using the quantum programming tool provided by projectQ. The NTT quantum circuit operates with three main functions. Each function performs an operation as much as a cycle(C), and Table 1 in Appendix shows the quantum resources used in each function. $fmul_1$ and $fmul_2$ perform multiplication on input and $zeta$. The difference between the two functions is that $fmul_2$ uses more quantum resources than $fmul_1$ because it has to determine the input sign expressed in two's complement. In $fmul_2$, the multi-controlled gate is used to determine the operation according to the sign of the input. The Montgomery reduce function uses the most quantum resources because it multiplies large numbers. Since the NTT sub function is a simple addition and subtraction operation for 32-bit qubits, it operates with the least amount of quantum resources. The quantum resource of each function is repeated as many as C, and a lot of resources are required to operate the NTT. Since the amount of quantum resources suggested only for Kyber's NTT operation is required, it is expected that a very large-scale quantum computer will be required to operate the entire cipher on a quantum computer.

5 Conclusion

In this paper, we propose a quantum circuit for NTT multiplication that is used to speed up polynomial multiplication in CRYSTALS-Kyber. The proposed quantum circuit was implemented as an optimization method to reduce quantum resources. We presented the quantum circuit pseudocode for NTT quantum computing and explained the operation of each function. Finally, we estimated quantum resources based on the proposed quantum circuit and analyze it. To the best of our knowledge, this is the first NTT quantum circuit to operate on the CRYSTALS-Kyber cryptographic algorithm. We expect that it can be used to evaluate the post-quantum security strength for CRYSTALS-Kyber.

6 Appendix

Table 1: Quantum resource for NTT function

Function	C	Quantum gates				Depth
		CCCNOT	Toffoli	CNOT	X	
$fmul_1$	128	-	48,576	97,943	1	146,488
$fmul_2$	768	97,024	195,564	33	2	292,592
Mont reduce	896	-	306,270	639,184	-	945,438
NTT sub	896	-	124	318	-	379

Algorithm 1 $fmul$ multiplication for $C = 1$

Input: $zeta$, r

for i=0 to $length(r)$ **do**
  $temp[i] \leftarrow$ CNOT($r[i]$, $temp[i]$)
end for

if $zeta \neq 1$ **then**
  if $zeta < 0$ and $input < 0$ **then**
    for i=0 to $-zeta+1$ **do**
      Dagger : $temp \leftarrow$**add**(r, $temp$)
    end for
  end if

  if $zeta < 0$ and $input > 0$ **then**
    for i=0 to $-zeta-1$ **do**
      $temp \leftarrow$**add**(r, $temp$)
    end for
  end if

  if $zeta \geq 0$ and $input > 0$ **then**
    for i=0 to $zeta-1$ **do**
      $temp \leftarrow$**add**(r, $temp$)
    end for
  end if

  if $zeta \geq 0$ and $input < 0$ **then**
    for i=0 to $zeta+1$ **do**
      Dagger : $temp \leftarrow$**add**(r, $temp$)
    end for
  end if
end if

return $temp$

Algorithm 2 $fmul$ multiplication for $C \geq 2$

Data: $zeta$, r, $check$(1-qubit)

$check \leftarrow$ CNOT(r[$length(r)$-1], $check$)
for (i=0 **to** $length(r)-1$) : $check \leftarrow$ CNOT(r[$length(r)$-1], $check$)

if $zeta \neq 1$ **then**
  if $zeta \geq 0$ **then**
    X($check$)
     if *check=1* **then**
      for (i=0 **to** $-zeta+1$) : **Dagger**: $temp \leftarrow$ **add**(r, $temp$)
    end
    X($check$)
     if *check=1* **then**
      for (i=0 **to** $-zeta-1$) : $temp \leftarrow$ **add**(r, $temp$)
    end
  end
else
  X($check$)
   if *check=1* **then**
    for (i=0 **to** $-zeta-1$) : $temp \leftarrow$ **add**(r, $temp$)
  end
  X($check$)
   if *check=1* **then**
    for (i=0 **to** $zeta+1$) : $temp$ **Dagger**: $\leftarrow$ **add**(r, $temp$)
  end
end
return $temp$

Algorithm 3 Montgomery reduce

Input: $a, temp_1, temp_2$
1: **for** i=0 to $-QINV$ **do**
2: **Dagger**: $tmp_1[0:16] \leftarrow$ **add**($a[0:16]$, $tmp_1[0:16]$)
3: **end for**

4: **for** i=0 to Q **do**
5: $tmp_2[0:32] \leftarrow$ **add**($tmp_1[0:32]$, $tmp_2[0:32]$)
6: **end for**

7: **Dagger**: $a[0:32] \leftarrow$ **add**($temp_2[0:32]$, $a[0:32]$)

return $a[16:32]$

References

[1] Ravi Anand, Arpita Maitra, and Sourav Mukhopadhyay. Grover on SIMON. *Quantum Information Processing*, 19(9):1–17, 2020.

[2] Anubhab Baksi, Kyungbae Jang, Gyeongju Song, Hwajeong Seo, and Zejun Xiang. Quantum implementation

and resource estimates for rectangle and knot. *Quantum Information Processing*, 20(12):1–24, 2021.

[3] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-kyber: a cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367. IEEE, 2018.

[4] Lily Chen, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. *Report on post-quantum cryptography*, volume 12. US Department of Commerce, National Institute of Standards and Technology . . . , 2016.

[5] Steven A Cuccaro, Thomas G Draper, Samuel A Kutin, and David Petrie Moulton. A new quantum ripple-carry addition circuit. *arXiv preprint quant-ph/0410184*, 2004.

[6] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[7] Zhenyu Huang and Siwei Sun. Synthesizing quantum circuits of aes with lower t-depth and less qubits. *Cryptology ePrint Archive*, 2022.

[8] Kyungbae Jang, Anubhab Baksi, Gyeongju Song, Hyunji Kim, Hwajeong Seo, and Anupam Chattopadhyay. Quantum analysis of aes. *Cryptology ePrint Archive*, 2022.

[9] Samuel Jaques, Michael Naehrig, Martin Roetteler, and Fernando Virdia. Implementing Grover oracles for quantum key search on AES and LowMC. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 280–310. Springer, 2020.

[10] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015.

[11] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[12] Gyeongju Song, Kyoungbae Jang, Hyunjun Kim, Siwoo Eum, Minjoo Sim, Hyunji Kim, Waikong Lee, and Hwajeong Seo. Speedy quantum circuit for grover's algorithm. *Applied Sciences*, 12(14):6870, 2022.

[13] Gyeongju Song, Kyungbae Jang, Hyunji Kim, Wai-Kong Lee, and Hwajeong Seo. Grover on Caesar and Vigenère ciphers. *IACR Cryptol. ePrint Arch.*, 2021:554, 2021.