IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 February 2023, accepted 23 February 2023, date of publication 3 March 2023, date of current version 8 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3252504

RESEARCH ARTICLE

Space and Time-Efficient Quantum Multiplier in Post Quantum Cryptography Era

DEDY SEPTONO CATUR PUTRANTO[1,3], (Member, IEEE), RINI WISNU WARDHANI[2], (Graduate Student Member, IEEE), HARASHTA TATIMMA LARASATI[2], (Graduate Student Member, IEEE), AND HOWON KIM[2], (Member, IEEE)

[1]IoT Research Center, Pusan National University, Busan 609735, South Korea
[2]School of Computer Science and Engineering, Pusan National University, Busan 609735, South Korea
[3]Blockchain Platform Research Center, Pusan National University, Busan 609735, South Korea

Corresponding author: Howon Kim (howonkim@pusan.ac.kr)

This work was supported in part by Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government Ministry of Science and Information and Communication Technology/ICT (MSIT) (No. 2019-0-00033, Study on Quantum Security Evaluation of Cryptography based on Computational Quantum Complexity, 50%); and in part by Energy Cloud R&D Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and Information and Communication Technology/ICT (MSIT) (NRF-2019M3F2A1073385).

ABSTRACT This paper examines the asymptotic performance of multiplication and the cost of quantum implementation for the Naive schoolbook, Karatsuba, and Toom-Cook methods in the classical and quantum cases and provides insights into multiplication roles in the post-quantum cryptography (PQC) era. Further, considering that the lattice-based PQC algorithm is based on polynomial multiplication algorithms, including the Toom-Cook 4-way multiplier as its fundamental building block, we propose a higher-degree multiplier, the Toom-Cook 8-way multiplier, which has the lowest asymptotic performance and implementation cost. Additionally, the designed multiplication will include additional sub-operations to complete the multiplication of large integers in order to prevent side-channel attacks. To design our Toom-Cook 8-way in detail, we employ detailed step computations such as splitting, evaluation, point-wise multiplication, interpolation, and recomposition, as well as several strategies to reduce space and time requirements. Existing asymptotic performance and quantum implementation cost multipliers are compared with our 2-way, 4-way, and 8-way Toom-Cook multiplier designs. Our Toom-Cook 8-way quantum multiplier has the lowest asymptotic performance analysis or qubit count in terms of space efficiency, with $n(\frac{15}{8})^{\frac{\log 15}{(2\log 15-\log 8)}\log_8 n}$ or asymptotically $\mathcal{O}(n^{1.245})$. The design has the lowest logical Toffoli counts bound at $112n^{\log_8 15} - 128n$ and Toffoli depth of $n(\frac{15}{8})^{1-\frac{\log 15}{(2\log 15-\log 8)}\log_8 n}$, asymptotically close to $\mathcal{O}(n^{1.0569})$, which corresponds to a space- and time-efficient multiplication.

INDEX TERMS Post quantum cryptography, quantum multiplication, Karatsuba, Toom-Cook 8-way, asymptotic performance.

I. INTRODUCTION

Numerous studies have attempted to explain arithmetic multiplication in classical or quantum computing environments from the Naïve Schoolbook onward, continuing through multiple multiplication development versions such as Karatsuba [1], [2], [3], Montgomery [4], and Toom-Cook [5], [6] [7], [8]. Notably, many other studies have been conducted in a variety of fields or from different perspectives on multiplication. The research was carried out in order to achieve low-complexity processing, optimal resource utilization, step improvements in multiplication, or efficient

The associate editor coordinating the review of this manuscript and approving it for publication was Chien-Ming Chen.

performance in modular arithmetic processing. In hardware classical computing and quantum processing, modular multiplication operations are also studied to meet the most effective and efficient computation in the prime fields $GF(p)$ or binary fields $GF(2^m)$.

Quantum computers, unlike classical computers, use the fundamentals of quantum mechanics to solve computational problems. The depth of a quantum circuit corresponds to its longest path. It represents the number of gates that qubits must sequentially pass through during an operation. In addition to the depth of the circuit, the complexity is also measured by the total number of qubits, generally corresponds to the circuit width, also known as the space requirement [2]. The researcher has been attempting to solve the critical problem of reducing the time complexity of the quantum gate structure and the width of quantum processing. Quantum circuits consisting of quantum gates, such as the NOT gate, CNOT gate, and Toffoli gate, are the fundamental building blocks for implementing various quantum algorithms in a quantum circuit. The depth and width of a quantum circuit for the arithmetic algorithm are related to complexity analysis.

The Open Quantum Safe (OQS) project at the National Institute of Standards and Technology (NIST) facilitates the development of a variety of quantum-resistant algorithms, such as code-based cryptography, multivariate quadratic equation-based cryptography, hash-based cryptography, isogeny-based cryptography, and lattice-based cryptography, which is particularly important as we begin moving into the post-quantum cryptography era [9]. The third phase of the NIST post-quantum cryptography (PQC) standardization process led to the identification of four candidate algorithms for standardization. NIST will recommend two principal algorithms for the majority of use cases: Crystals-Kyber (key establishment) and Crystals-Dilithium (digital signatures) [9]. In addition, Falcon and SPHINCS+ will be standardized [9].

Crystal-Kyber, NTRU, and SABER are examples of lattice-based post-quantum encryption; they rely on polynomial multiplication algorithms like Toom-Cook and the Number Theoretic Transform (NTT) [10]. Further investigation in their research revealed that multiplication can be exploited in multiplication-based attacks. Mujdei et al. [10] demonstrated that side-channel analysis (SCA) can be performed as a practical experiment on real-world side-channel measurements, allowing the secret key to be extracted from lattice-based post-quantum key encapsulation mechanisms. Their investigation shows that the polynomial multiplication method used can significantly affect the attack's temporal complexity [10].

Given how important polynomial multiplication algorithms like Toom-Cook and NTT are to lattice-based post-quantum cryptography, this paper proposes a higher-degree Toom-Cook multiplier, i.e., Toom-Cook 8-way multiplier, which achieves the best asymptotic performance and implementation cost. We propose a multiplier that has a lower space-and-time complexity and can be used to perform efficient quantum cryptanalysis. In addition, the intended multiplication is anticipated to be defended against the multiplication-based attack in a lattice-based technique by producing more iterations to lower the peak in the correlation power analysis (CPA) in SCA. Keeping this in mind, the basic operations that make up the PQC algorithm in the post-quantum era may be more secure and meet PQC security requirements if they are more effective and less easy to exploit.

The following is a summary of the paper's contribution:

1) This study provides a comprehensive analysis of existing multiplication algorithms by presenting quantum asymptotic performance comparison results and quantum implementation costs for the popular multiplication algorithms (i.e., Naive schoolbook, Karatsuba, and Toom-Cook algorithms), in terms of qubit count, Toffoli depth, and Toffoli count.
2) We investigate the significance of multiplication in the PQC era, including its role as part of the PQC algorithm, multiplication exploitation in the CPA-based approach in SCA, and as a constructor in the cryptanalysis circuit.
3) The proposed Toom-Cook 8-way design multiplier is expected to make the multiplication process go through more multiplication sub-operations, which will later affect the results of power analysis using the CPA-based method to avoid multiplication-based attacks. We describe our Toom-Cook 8-way design with detailed computation steps like splitting, evaluation, recursion, interpolation, and recomposition. Further, we assess the Toom-Cook 8-way multiplier design and provide multiplication performance in a quantum environment to determine the complexity of the Toom-Cook 8-way multiplier.
4) We compare the design multiplier to the Naive schoolbook, Karatsuba, and existing Toom-Cook complexity analyses to determine the lowest asymptotic performance multiplication and cost in the quantum case, including our 2-way, 4-way, and 8-way Toom-Cook multipliers. Comparing the qubit count, the Toffoli depth, and the Toffoli count, the Toom-Cook 8-way has the most asymptotically efficient resource utilization for multiplication. The Toom-Cook 8-way Quantum Multiplier has the smallest asymptotic performance analysis in terms of qubit count $n(\frac{15}{8})^{\frac{\log 15}{(2\log 15-\log 8)}\log_8 n}$, with a value close to $\mathcal{O}(n^{1.245})$. In terms of temporal complexity, the design has the lowest logical Toffoli count that bounds to $112n^{\log_8 15} - 128n$ and Toffoli depth of $n(\frac{15}{8})^{1-\frac{\log 15}{(2\log 15-\log 8)}\log_8 n} \approx n^{1.0569}$.

II. COMPLEXITY OF VARIOUS MULTIPLICATION ALGORITHMS

Schoolbook multiplication is the naive method of multiplying numbers by first multiplying the multiplicand by each digit of the operand and then summing the results after performing

any necessary shifts. The time complexity of a multiplication arithmetic algorithm is a theoretical measurement for assessing how long it will take to run in practice, and classically, space complexity is a theoretical measurement related to how much memory an algorithm needs to run. The schoolbook multiplication algorithm has an $\mathcal{O}(n^2)$ level of complexity, which indicates that it takes approximately n^2 operations to multiply two $n-$digit numbers. Consequently, the computation time practically explodes (i.e., at asymptotic $\mathcal{O}(n^2)$) for large values of n.

Karatsuba and Ofman [11], often referred to as Karatsuba, is an algorithm for multiplying subsquare polynomials. The Karatsuba method has seen significant advances in speeding up large number multiplication instead of the schoolbook's conventional ways. It has numerous variations that can be implemented on any hardware or quantum-based computation. Certain Karatsuba multiplication variations implemented in quantum circuits, such as in [1], [3], and [12], have been devised to reduce processing complexity. Karatsuba multiplication can also be considered as the foundation of Toom-Cook-based multiplication, in which the operand is derived from some polynomials, an evaluation point is chosen, then point evaluation is interpolated.

Toom-Cook multiplier algorithms are efficient methods for multiplying subquadratic polynomials or long integers [6]. Given two large integers x and y, the Toom-Cook algorithm splits them into k smaller parts of length l. Then, the sub-multiplications are done using Toom-Cook multiplication in a recursive way until we can use a different method for the last step of recursion or we reach the target multiplier. Subsequently, the Toom-Cook k-way, introduced by Andrei L. Toom in 1963 [5], [13], generalizes Karatsuba's concept by dividing $rk-$coefficient polynomials into k parts, performing $r-$coefficient multiplication, and then combining the results [10]. Before using the schoolbook for sub-operations multiplication, it is often necessary to multiply the sub-polynomials by applying a number of Karatsuba-layers.

Naive and Karatsuba multipliers may be identified as similar Toom-Cook family algorithms, as mentioned in [6], because they divide the operand's value into 1 and 2 parts, respectively. Toom-Cook multiplier algorithms, in practice, we use the degree 2 (Karatsuba), 3, and 4 for variant versions, which are efficient subquadratic polynomial or long integer multiplication methods [6]. Bodrato and Zanoni generalized the original Toom-Cook family in [14] by considering unbalanced operands —that is, polynomials of varying degrees known as Toom$-(k + 1/2)$ methods [6].

The complexity of Karatsuba with $\mathcal{O}(n^{\log_2(3)})$ is comparatively lower than that of the schoolbook algorithm; more specifically, its power is $\log_2 3 \equiv 1.585$ as opposed to quadratic in the schoolbook case.

In terms of multiplication research for the quantum case, Figure 1 depicts a quantum implementation of an improved multiplication Karatsuba with the parameter $m = 8$, where m represents the highest degree in the polynomial that is being multiplied [15]. In addition, Dutta et al., [7]presented the quantum circuit for Toom-Cook integer multiplication. In particular, the authors design the circuit for the Toom-Cook 2.5-way multiplication and approximate the number of Toffoli gates and qubits by looking at the recursive tree structure of their method. Other research, conducted by Larasati et al., [8] referring to the classical implementation in [14], investigated the Toom-Cook 3-way multiplication design of the quantum circuit, giving an asymptotically lower depth than the Toom-Cook 2.5-way circuit. The difficulty in developing higher-degree Toom-Cook multiplication involving odd numbers is that there is a resource bottleneck because it still involves nontrivial-division operations [8].

Other multipliers exist besides the Schoolbook, Karatsuba, and Toom-Cook; however, to the best of our knowledge, they have few quantum implementation equivalents. As the Toom-Cook-based multiplier is a crucial component of our proposed multiplier design, this paper expands only on the Naive Schoolbook, Karatsuba, and Toom-Cook multiplier methods and their asymptotic performance analysis.

III. MULTIPLICATION ROLE IN POST QUANTUM CRYPTOGRAPHY ERA

The consistent advances in quantum computing have led to concerns that the widespread use of public-key cryptosystems could be compromised if large-scale quantum computers were to be available in the future. In particular, this can lead to the success of quantum factoring, enabling the obtaining of key parameters of public-key cryptosystems such as elliptic curve-based cryptography and digital signatures. As a direct result, there has been an increase in research into the development of public-key cryptosystems that are secure against attackers employing classical and quantum computers. PQC is the name for this area of study, along with quantum-resistant cryptography [9].

A. TOOM-COOK IN LATTICE-BASED ALGORITHM

The most obvious application of multiplication in the PQC era is in the development of PQC algorithms. NIST has recently settled algorithms for standardization after a rigorous assessment in the third stage of the NIST PQC standardization process. Crystals-Kyber [16] for the public-key establishment and Crystals-Dilithium [17] are the two fundamental algorithms that NIST recommends for the vast majority of use cases, including digital signatures. Lattice-based cryptography, which is predicted to be the most effective and quantum-safe, provides the necessary solution in the era of PQC and appears to be the most rapid implementation as in [18], [19], [20], and [21]. Dilithium, Falcon, FrodoKEM, Kyber, NTRU, NTRU Prime, and Saber are seven of the fifteen candidates in the NIST third round that use lattice-based cryptography [9]. Lattice-based cryptosystems typically rely on either the NTT ($\mathcal{O}(n \log n)$) [22] or the Toom-Cook/Karatsuba ($\mathcal{O}(n^{1+\epsilon}), 0 < \epsilon < 1$) [5], [11], [13] for fast multiplication of polynomials with n coefficients [10]. A comparison of the algorithm runtime behavior and memory consumption of

FIGURE 1. Karatsuba's improvement multiplication with the highest degree polynomial is m = 8. On a 64-bit Windows 10 Pro (Intel i7-8700, six-core CPU) with 64 GB of RAM and Python 3.9.7, the quantum circuit yields from the Qiskit simulation.

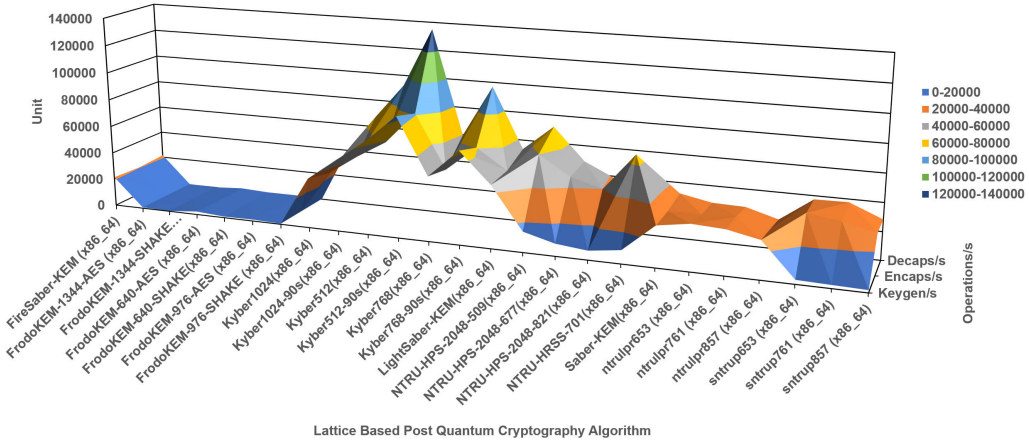

FIGURE 2. Runtime analysis of Open Quantum Safe lattice-based cryptographic algorithms (key encapsulation mechanisms). We redraw the results of the runtime analysis of a secure quantum lattice-based cryptographic algorithm (key encapsulation mechanism) based on a comparative analysis of the algorithm's runtime behavior and memory consumption data in [10].

Open Quantum Safe lattice-based cryptographic algorithms (key encapsulation mechanisms) is depicted in Figure 2 [10].

Niasar's research, in [21], led to an architecture that utilizes NTT to speed up polynomial multiplication. This was proposed as a solution to Kyber's adequate computing time on traditional hardware [21]. Lattice-based post-quantum encryption is dependent on polynomial multiplication algorithms, such as the Toom-Cook 4-way algorithm [10].

B. MULTIPLICATION-BASED ATTACKS

Side-channel analysis (SCA) and fault-injection attacks pose serious threats to cryptographic implementations [23]. SCA is mitigated by techniques that conceal or mask key-dependent information, whereas resistance to fault-injection attacks can be achieved by adding redundancy for immediate error detection [24]. Concerning the fact that side-channel attacks are nowadays serious when implementing cryptographic algorithms, powerful ways for gaining information about the secret key as well as various countermeasures against such attacks have been recently developed [25].

Some research investigates attack countermeasures or the efficacy of the attacks in order to prevent the threat. Regazzoni et al.'s research investigate a device's resistance to power attacks in order to protect it from fault injection attacks when fault detection circuitry is added [26]. They attempt to put modified devices to the test against attacks based on power analysis. A study in [23] examines the impact of a fault-detection (FD) scheme on the robustness of a full AES

implementation against correlation power analysis (CPA). For SHA-3 third-round finalists, the [27] study introduced an efficient and effective error detection scheme.

Physical attacks such as SCA and fault analysis (FA) can be used to recover the secret key used in a cryptosystem [28]. A side-channel attack is a type of computer security attack that does not rely on flaws in the design of a computer protocol or algorithm, but rather on additional information that can be gleaned from how the protocol or algorithm is employed. Utilizing additional data such as timing, power consumption, electromagnetic leaks, and sounds can facilitate side-channel attacks. SCA on cryptographic algorithm implementations has been carried out in recent years, demonstrating how to obtain the secret key [26].

Several studies demonstrate that arithmetic such as multiplication is also a concern as an object in SCA. Whelan and Scott [29], proposed one of the first articles describing side-channel attacks against large characteristic field pairings [29]. They propose the calculation strategy for the CPA portion to calculate the correlations between the hypothetical outputs of the arithmetic operation x x k and the leakage traces for all possible keys k. The research of [26] demonstrates that both S−box analysis and multiplication can be used in a side-channel attack against the AES and Kasumi ciphers.

C. MULTIPLICATION EXPLOITATION FOR CPA SIDE CHANNEL ANALYSIS

Side-channel data, such as power consumption, electromagnetic (EM) radiation, and execution time, can be used to get at sensitive data [30]. Side-channel attacks are passive physical assaults in which the attacker obtains side-channel data produced by the implementation inadvertently [10]. One of the most effective is correlation power analysis (CPA), which takes advantage of the relationship between a device's power consumption and the data it is processing, such as power fluctuations caused by multiplication.

Multiplication exploitation in CPA side-channel analysis attacks is a major concern when implementing cryptographic algorithms, as in a practical implementation, arithmetic multiplication is almost used as a sub-operation multiplier, leaving cryptographic algorithms susceptible to physical attack exploitation [10]. Mujdei et al. compared the complexity of attacking various multiplication schemes, multiplication techniques, and parameter selections utilizing CPA, a technique first introduced by Brier et al. in their 2004 paper [31]. Using side-channel measurements, they demonstrated that their method decrypts all lattice-based post-quantum key encapsulation schemes. In addition, they demonstrate that the time complexity of an attack can be drastically altered through the use of polynomial multiplication [10]. So, side-channel analysis techniques based on correlation power analysis can be used to attack the existing polynomial multiplication strategies used in lattice-based post-quantum key encapsulation mechanisms. Figure 3 clearly shows a multiplication-based attack on the post-quantum algorithm using Mujdei et al.'s findings about the variance plot of 500 schoolbook multiplication traces with 72 peaks for inntruhps4096821 [10].

D. EFFICIENT MULTIPLIER FOR QUANTUM CRYPTANALYSIS

PQC refers to cryptographic methods, typically algorithms for public key encapsulation, that are believed to be secure against quantum computer attacks. PQC focuses on updating math-based algorithms and standards to prepare for the quantum computing era. Currently, we require efficient mathematics not only to construct PQC algorithms that are resistant to SCA but also in the form of quantum circuits that can be used to construct cryptanalysis circuits, which can later be used to demonstrate the strength of an algorithm. Multiplication of integers is a fundamental operation on a conventional computer. In quantum computing, integer multiplication is a crucial operation that is fundamental to executing Shor's algorithm for factoring integers [32].

Utilizing Shor's algorithm to solve the factoring problem in RSA or ECC cryptographic systems have been the primary focus of early quantum computing cryptanalysis research. In general, the objective is to design a circuit that efficiently computes the modular exponential function $|X\rangle \rightarrow |a^x \bmod N\rangle$, enabling its solution via a modular multiplication sequence. Several studies, such as [33], [34], and [35], have introduced methods for achieving this objective, where they perform multiplication by a constant using a modular arithmetic method and a single quantum input. By utilizing the unique properties of constants to design more idealized circuits, [36] and [37] developed a more sophisticated solution. Quantum-based cryptanalysis requires basic arithmetic, including multiplication, for all of these investigations.

To accelerate cryptanalysis, a space- and time-efficient basic arithmetic constructor is required. Further research into quantum computing environments, such as that conducted by Roche [38], Parent et al. [2], Gidney [39], Banegas et al. [40], and Putranto et al. [15], focuses on reducing the temporal or spatial complexity of cryptanalysis implementation. Banegas et al. research in [40] is an example of research that uses space and time-efficient quantum multiplication to enhance cryptanalysis through the development of multiplication research from [3]. Future predictions regarding the performance of quantum computers to solve classical public key cryptography and PQC algorithms can be greatly influenced by the efficiency of the underlying basic arithmetic, in this case, multiplication.

E. SPACE AND TIME-EFFICIENT MULTIPLIER WITH HIGHER-DEGREE TOOM-COOK-BASED MULTIPLIER

In this paper, we disassemble the design of the classical Toom-Cook multiplier and redesign it so that it can also be used as quantum circuits, which can later be implemented in actual quantum hardware systems. One of the main concerns

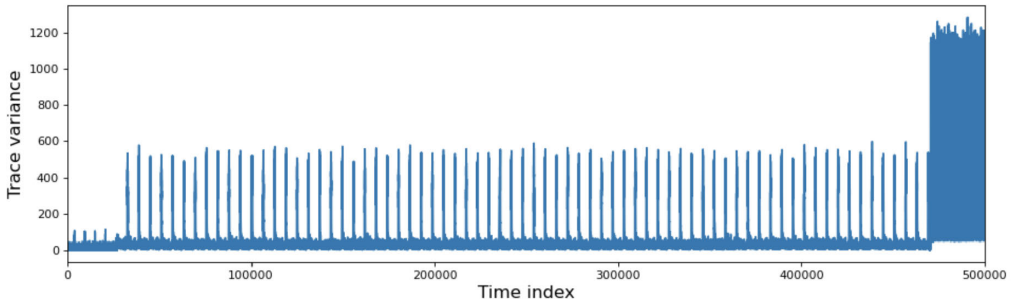

FIGURE 3. Variance plot of 500 traces of schoolbook multiplication in ntruhps4096821, showing 72 peaks. The result CPA is for the case of ntruhps4096821 with size 1728 NTT, schoolbook threshold degree 3, and eight 3C–3 schoolbook multiplications per group. [10].

is how to design multiplication to utilize the least amount of space and time. This is due to the fact that multiplication is a fundamental mathematical operation that is frequently employed and can be an integral component of complex mathematics and cryptanalysis.

According to Mujdei et al. [10], lattice-based post-quantum cryptography is built on polynomial multiplication algorithms such as Toom-Cook and the NTT. Toom-Cook-based versions of Saber and NTRU use $k = 4$ with evaluation points $x_1 = 0$, $x_2 = 1$, $x_3 = -1$, $x_4 = 2$, $x_5 = -2$, $x_6 = 3$, and $x_7 = \infty$ [10]. Mujdei's research not only shows that Toom-Cook is used in lattice-based PQC algorithms, but it also shows that the multiplication of Toom-Cook-based polynomials is straightforward to attack [10].

As the fundamental building blocks of lattice-based post-quantum encryption, polynomial multiplications (e.g., Toom-Cook and NTT) further divide the resulting sub-polynomial [10]. Saber further splits the resulting sub-polynomials into two Karatsuba-layers, after which a 16-coefficient schoolbook is performed [10]. The structure of all NTRU-versions is similar, but with four Karatsuba-layers (except ntruhps2048509, which has three) and different schoolbook thresholds [10]. By experiment, Mujdei et al. analyzed whether the schoolbook sub-operation in processing 3-way and 4-way Toom-Cook in the lattice-based PQC algorithm resulted in CPA peaks. Taking into account the peak results derived from the schoolbook multiplication sub-operation, as depicted in Figure 3 by Mujdei et al., it is hypothesized that Toom-Cook with a high degree will require more schoolbook sub-operations to multiply numbers and is expected to produce more peaks. If a design with high-order multiplication is implemented, side-channel attacks will have a greater number of peaks, making them more difficult to analyze. In the end, the designed multiplier should have more possible sub-operations so that more peaks can be made while still using space and time efficiently.

IV. DESIGN OF QUANTUM TOOM-COOK 8-WAY MULTIPLIER

Having observed that Toom-Cook is used as a polynomial-based multiplication algorithm on a lattice-based algorithm in the PQC era, we devised a high-degree Toom-Cook-based multiplication, which should provide a lower depth than the other method of multiplication and prevent multiplication-based exploitation. We take a close look at multiplication and come up with a new multiplier based on the work of Zanoni et al. [6], Dutta et al. [7], and Larasati et al. [8], among others.

Zanoni et al. offer a degree 7 implementation of a balanced Toom-Cook 8-way for integer multiplication and squaring. The design research methodology used by Zanoni to create the Toom-Cook 8-way. In classical computing, balance is achieved without the use of non-trivial division [6]. By examining the algorithm's recursive tree, Dutta et al. provide a comprehensive explanation of the Toom-Cook 2.5-way method for obtaining a limit on the number of Toffoli gates and qubits in quantum computing [7].

Larasati et al. [8] elaborate on the Toom-Cook 3-way by citing Bodrato [14]. Both researchers continue to use the division gate, and the objective is to reduce the number of operations, particularly nontrivial ones. Some of Larasati et al.'s strategies emphasize the use of division gates, which have resulted in a maximum of one exact division by three circuits per iteration [8]. In addition, the cost of the remaining division was reduced by utilizing the circuit's unique property and replacing it with a constant multiplication by reciprocal circuit and the corresponding swap operations [8]. The research of Larasati et al. [8] demonstrates that the $k-$way Toom-Cook method, which employs higher-order polynomial interpolation, can have lower asymptotic complexity than other methods such as Toom-2.5 in terms of both the number of qubits and the depth of the Toffoli tree. However, it also requires a greater quantity of Toffoli gates, primarily due to the manner in which it is divided. Also, research has demonstrated that despite the fact that higher-order methods may be more efficient, it can be difficult to find an efficient way to implement the division operation, which is a crucial component of the $k-$way Toom-Cook method.

The designed Toom-Cook 8-way quantum multiplier will implement the concept of balanced Toom-Cook 8-way classical computation as described in Zanoni et al.'s strategy to avoid using non-trivial division when building the

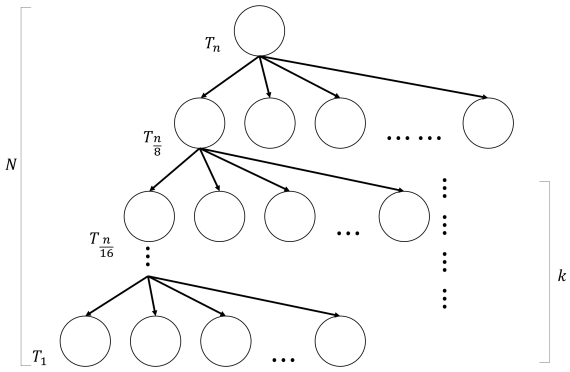

FIGURE 4. The Toom-Cook 8-way Multiplication Recursion Tree Structure, where T stands for the Toom-Cook 8-way Multiplication, *n* and *N*, respectively, stand for the bit length for each level and the overall depth of the tree.

balanced Toom-Cook 8-way in classical computing. To maximize space and time efficiency, the multiplier will employ several strategies, with an emphasis on not using the division gate. Despite using the division function in the Toom Cook 8-way design, we only use the copy and add functions, which only require the CNOT gate. Even though the Toom-Cook 8-way has a higher degree, the initial resource requirements for the number of qubits, the depth of the Toffoli, and the number of Toffoli will be lower. We do not use a division gate when designing space and time-efficient multiplication, and we demonstrate that the use of gates and the depth count of multiplication circuit resources will be reduced.

A. TOOM-COOK 8-WAY COMPUTATION STEPS

Existing quantum research on the degree of Toom-Cook methods, which provide faster execution in large number multiplication than schoolbook and Karatsuba, has multiple forms, including 2.5-way, 3-way, and 4-way. The $k-$way Toom Cook method is a multi-point polynomial multiplication algorithm that uses interpolation to multiply two large polynomials efficiently by decomposing them into smaller polynomials and recursively applying the same technique to these smaller polynomials until they are small enough to be multiplied using the standard method. The sub-operations of multiplication can then be computed recursively using Toom-Cook multiplication, and so on. The Toom-Cook 8-way multiplication recursion tree structure is shown in Figure 4. The Toom-Cook 8-way method divides a large integer number into 16 smaller multiplications, then operates the calculation for each part. The calculations take into account the quantum circuit's resource properties, such as reversibility, which necessitate uncomputation in order to eliminate garbage outputs. In quantum circuit research, it has also been discovered that quantum computing typically requires more resources than classical computing. Even in the same study in the quantum circuit field, there are examples of high resources for the number of Toffoli on Toom-Cook 3-way, which is greater than the number of Toffoli on Toom-Cook 2.5-way, as stated in [8].

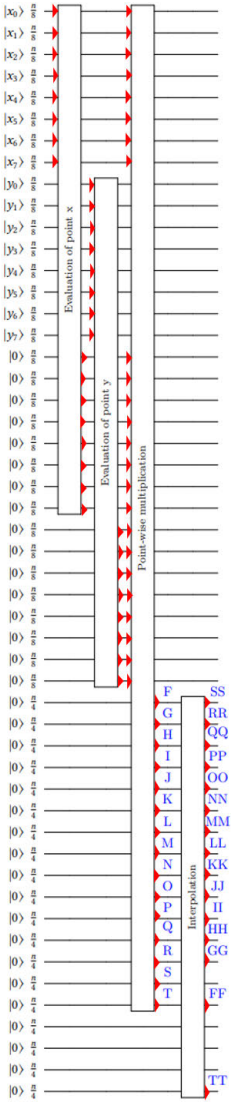

FIGURE 5. Overall Quantum Circuit for Toom-Cook 8-way Multiplication. The function block boxes represent the steps that comprise the Toom-Cook 8-way quantum circuit. In the Quantum Circuit for Toom-Cook 8-way multiplication, the red triangles at each function block indicate the input and output of each corresponding operation. A notation symbol represents the quantum state of the input, and each line represents a required register in the quantum circuit. Triangles on the left of a block indicate its input entry point. Triangles represent the output location on the right side. For simplicity's sake, the ancillary registers are not displayed.

Zanoni's research in classical balanced Toom-Cook 8-way for long integers multiplication indicates five steps to perform the Toom-Cook k-way algorithm for a natural number such as splitting, evaluation, recursion, interpolation, and recomposition [6]. In order to provide a concise description of the method, the values to be multiplied, also known as the input operands, are denoted by x and y. The symbol x denotes the full-digit input, $x_0, x_1, x_{-1}, x_{-2}, \ldots$ represents the split input, whereas $x(0), x(1), x(-1), x(-2), \ldots$ represents the outcome of evaluating x for the specified evaluation points. Following is the procedure for implementing quantum Toom-Cook 8-way Multiplication:

1) As demonstrated by Equations 1 and 2, each input, in this case, x and y, is divided into eight smaller pieces of length $\frac{n}{8}$. The radix j in the equations can be calculated beforehand using Equation 3.

$$x = x_7 s^{7j} + x_6 s^{6j} + x_5 s^{5j} + x_4 s^{4j} + x_3 s^{3j} + x_2 s^{2j} + x_1 s^{j} + x_0 \quad (1)$$

$$y = y_7 s^{7j} + y_6 s^{6j} + y_5 s^{5j} + y_4 s^{4j} + y_3 s^{3j} + y_2 s^{2j} + y_1 s^{j} + y_0 \quad (2)$$

$$j = max\left\{\left\lfloor \frac{\lceil \log_2 x \rceil}{8} \right\rfloor, \left\lfloor \frac{\lceil \log_2 y \rceil}{8} \right\rfloor\right\} \quad (3)$$

2) We employ $x_1 = 0$, $x_2 = 1$, $x_3 = -1$, $x_4 = 2$, $x_5 = -2$, $x_6 = 4$, $x_7 = -4$, $x_8 = 8$, $x_9 = -8$, $x_{10} = 16$, $x_{11} = -16$, $x_{12} = 32$, $x_{13} = -32$, $x_{14} = -64$, $x_{15} = \infty$ to obtain $x(0)$, $x(1)$, $x(-1)$, $x(2)$, $x(-2)$, $x(4)$, $x(-4)$, $x(8)$, $x(-8)$, $x(16)$, $x(-16)$, $x(32)$, $x(-32)$, $x(-64)$, and $x(\infty)$ for the evaluating points x and y, each of the 15 predefined evaluation points. In this paper, we did not provide the detailed equation for the result evaluation point $x(0)$, $x(1)$, $x(-1)$, $x(2)$, $x(-2)$, $x(4)$, $x(-4)$, $x(8)$, $x(-8)$, $x(16)$, $x(-16)$, $x(32)$, $x(-32)$, $x(-64)$, $x(\infty)$; but, it can be recognized from the evaluation multiplication equation, Equation 4.

3) One round of non-recursive point-wise multiplication Toom-Cook 8-way multiplication employs 16 multiplications with smaller bit lengths. To multiply each component of $x(0)$, $x(1)$, $x(-1)$, $x(2)$, $x(-2)$, $x(4)$, $x(-4)$, $x(8)$, $x(-8)$, $x(16)$, $x(-16)$, $x(32)$, $x(-32)$, $x(-64)$, and $x(\infty)$, the result is presented in Equation 4, indicated as the notions F, G, H, I, J, K, L, M, N, O, P, Q, R, S, and T.

4) Interpolation can be modeled in a matrix as it is the opposite of a point multiplication result, as shown in Equation 5. Note that, in this process, we use an inverse matrix from coefficient sub-multiplication $(k_0 \ldots k_{14})$ in Equation 4; for simplicity, we describe the inverse matrix as in Equation 5.

$$F = x_0 y_0$$

$$G = (x_7 + x_6 + x_5 + x_4 + x_3 + x_2 + x_1 + x_0)(y_7 + y_6 + y_5 + y_4 + y_3 + y_2 + y_1 + y_0)$$

$$H = (-x_7 + x_6 - x_5 + x_4 - x_3 + x_2 - x_1 + x_0)(-y_7 + y_6 - y_5 + y_4 - y_3 + y_2 - y_1 + y_0)$$

$$I = (128x_7 + 64x_6 + 32x_5 + 16x_4 + 8x_3 + 4x_2 + 2x_1 + x_0)(128y_7 + 64y_6 + 32y_5 + 16y_4 + 8y_3 + 4y_2 + 2y_1 + y_0)$$

$$J = (-128x_7 + 64x_6 - 32x_5 + 16x_4 - 8x_3 + 4x_2 - 2x_1 + x_0)(-128y_7 + 64y_6 - 32y_5 + 16y_4 - 8y_3 + 4y_2 - 2y_1 + y_0)$$

$$K = (16384x_7 + 4096x_6 + 1024x_5 + 256x_4 + 64x_3 + 16x_2 + 4x_1 + x_0)(16384y_7 + 4096y_6 + 1024y_5 + 256y_4 + 64y_3 + 16y_2 + 4y_1 + x_0)$$

$$L = (-16384x_7 + 4096x_6 - 1024x_5 + 256x_4 - 64x_3 + 16x_2 - 4x_1 + x_0)(-16384y_7 + 4096y_6 - 1024y_5 + 256y_4 - 64y_3 + 16y_2 - 4y_1 + x_0)$$

$$M = (2097152x_7 + 262144x_6 + 32768x_5 + 4096x_4 + 512x_3 + 64x_2 + 8x_1 + x_0)(2097152y_7 + 262144y_6 + 32768y_5 + 4096y_4 + 512y_3 + 64y_2 + 8y_1 + y_0)$$

$$N = (-2097152x_7 + 262144x_6 - 32768x_5 + 4096x_4 - 512x_3 + 64x_2 - 8x_1 + x_0)(-2097152y_7 + 262144y_6 - 32768y_5 + 4096y_4 - 512y_3 + 64y_2 - 8y_1 + y_0)$$

$$O = (268435456x_7 + 16777216x_6 + 1048576x_5 + 65536x_4 + 4096x_3 + 256x_2 + 16x_1 + x_0)(268435456y_7 + 16777216y_6 + 1048576y_5 + 65536y_4 + 4096y_3 + 256y_2 + 16y_1 + y_0)$$

$$P = (-268435456x_7 + 16777216x_6 - 1048576x_5 + 65536x_4 - 4096x_3 + 256x_2 - 16x_1 + x_0)(-268435456y_7 + 16777216y_6 - 1048576y_5 + 65536y_4 - 4096y_3 + 256y_2 - 16y_1 + y_0)$$

$$Q = (34359738368x_7 + 1073741824x_6 + 33554432x_5 + 1048576x_4 + 32768x_3 + 1024x_2 + 32x_1 + x_0)(34359738368y_7 + 1073741824y_6 + 33554432y_5 + 1048576y_4 + 32768y_3 + 1024y_2 + 32y_1 + y_0)$$

$$R = (-34359738368x_7 + 1073741824x_6 - 33554432x_5 + 1048576x_4 - 32768x_3 + 1024x_2 - 32x_1 + x_0)(-34359738368y_7 + 1073741824y_6 - 33554432y_5 + 1048576y_4 - 32768y_3 + 1024y_2 - 32y_1 + y_0)$$

$$S = (-4398046511104x_7 + 6871946736x_6 - 1073741824x_5 + 16777216x_4 - 262144x_3 + 4096x_2 - 64x_1 + x_0)(-4398046511104y_7 + 6871946736y_6 - 1073741824y_5 + 16777216y_4 - 262144y_3 + 4096y_2 - 64y_1 + y_0)$$

$$T = x_7 y_7 \tag{4}$$

$$\begin{pmatrix} TT \\ SS \\ RR \\ QQ \\ PP \\ OO \\ NN \\ MM \\ LL \\ KK \\ JJ \\ II \\ HH \\ GG \\ FF \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} F \\ G \\ H \\ I \\ J \\ K \\ L \\ M \\ N \\ O \\ P \\ Q \\ R \\ S \\ T \end{pmatrix} \tag{5}$$

5) The recomposition from the interpolation result is indicated as TT, SS, RR, QQ, PP, OO, NN, MM, LL, KK, JJ, II, HH, GG, and FF in Equation 6 below. The final product of Toom-Cook 8-way multiplication is the xy Equation.

$$\begin{aligned} xy = {} & FF2^{14j} + GG2^{13j} + HH2^{12j} + II2^{11j} + JJ2^{10j} \\ & + KK2^{9j} + LL2^{8j} + MM2^{7j} + NN2^{6j} + OO2^{5j} \\ & + PP2^{4j} + QQ2^{3j} + RR2^{2j} + SS2^{j} + TT \end{aligned} \tag{6}$$

B. TOOM-COOK 8-WAY MULTIPLICATION SCHEDULING

We employ effective scheduling or sequencing, which entails merging similar processes with value overlapping, in order to reduce down on the number of operations that need to be performed. Equation 7−9 shows the best way to do Toom-Cook 8-way multiplication based on this idea, which was also used by [6], [7], and [8], in earlier studies to lower the cost of Toom-Cook based multiplication. It is worth noting that common quantum techniques still rely on two-input operations, unlike their classical counterparts, which can operate on multiple values simultaneously.

Let:

$$\mathbf{i}_1 = G + H \tag{7}$$

$$\mathbf{i}_2 = I + J \tag{8}$$

$$\mathbf{i}_3 = K + L \tag{9}$$

With the substitution of Equation 7−9, the formulation of Toom-Cook eight-way interpolation with the result of inverse matrices Equation 5, denoted by coefficient $k_0 \ldots k_{14}$, can be expressed as Equation 10 -24:

$$FF = T \tag{10}$$

$$\begin{aligned} GG = {} & k_0F + k_1G + k_2H + k_3I + k_4J + k_5K \\ & + k_6L + k_7M + k_8N + k_9O + k_{10}P \\ & + k_{11}Q + k_{12}R + k_{13}S + k_{14}T \end{aligned} \tag{11}$$

$$\begin{aligned} HH = {} & k_0F + k_1G + k_2H + k_3I + k_4J + k_5K \\ & + k_6L + k_7M + k_8N + k_9O + k_{10}P \\ & + k_{11}Q + k_{12}R + k_{13}S + k_{14}T \end{aligned} \tag{12}$$

$$\begin{aligned} II = {} & k_0F + k_1G + k_2H + k_3I + k_4J + k_5K \\ & + k_6L + k_7M + k_8N + k_9O + k_{10}P \\ & + k_{11}Q + k_{12}R + k_{13}S + k_{14}T \end{aligned} \tag{13}$$

$$\begin{aligned} JJ = {} & k_0F + k_1G + k_2H + k_3I + k_4J + k_5K \\ & + k_6L + k_7M + k_8N + k_9O + k_{10}P \\ & + k_{11}Q + k_{12}R + k_{13}S + k_{14}T \end{aligned} \tag{14}$$

$$\begin{aligned} KK = {} & k_0F + k_1G + k_2H + k_3I + k_4J + k_5K \\ & + k_6L + k_7M + k_8N + k_9O + k_{10}P \\ & + k_{11}Q + k_{12}R + k_{13}S + k_{14}T \end{aligned} \tag{15}$$

$$\begin{aligned} LL = {} & k_0F + k_1i_1 + k_2I + k_3J + k_4K + k_5L \\ & + k_6M + k_7N + k_8O + k_9P + k_{10}Q \\ & + k_{11}R + k_{12}S + k_{13}T \end{aligned} \tag{16}$$

$$\begin{aligned} MM = {} & k_0F + k_1G + k_2H + k_3I + k_4J + k_5K \\ & + k_6L + k_7M + k_8N + k_9O + k_{10}P \\ & + k_{11}Q + k_{12}R + k_{13}S + k_{14}T \end{aligned} \tag{17}$$

$$\begin{aligned} NN = {} & k_0F + k_1i_1 + k_2i_2 + k_3i_3 + k_4M + k_5N \\ & + k_6O + k_7P + k_8Q + k_9R + k_{10}S + k_{11}T \end{aligned} \tag{18}$$

$$\begin{aligned} OO = {} & k_0F + k_1G + k_2H + k_3I + k_4J + k_5K \\ & + k_6L + k_7M + k_8N + k_9O + k_{10}P \\ & + k_{11}Q + k_{12}R + k_{13}S + k_{14}T \end{aligned} \tag{19}$$

$$\begin{aligned} PP = {} & k_0F + k_1G + k_2H + k_3I + k_4J + k_5i_3 \\ & + k_6M + k_8N + k_7O + k_9P \\ & + k_{10}Q + k_{11}R + k_{12}S + k_{13}T \end{aligned} \tag{20}$$

$$\begin{aligned} QQ = {} & k_0F + k_1G + k_2H + k_3I + k_4J + k_5K \\ & + k_6L + k_7M + k_8N + k_9O + k_{10}P \\ & + k_{11}Q + k_{12}R + k_{13}S + k_{14}T \end{aligned} \tag{21}$$

$$\begin{aligned} RR = {} & k_0F + k_1G + k_2H + k_3I + k_4J + k_5K \\ & + k_6L + k_7M + k_8N + k_9O + k_{10}P \\ & + k_{11}Q + k_{12}R + k_{13}S + k_{14}T \end{aligned} \tag{22}$$

$$\begin{aligned} SS = {} & k_0F + k_1G + k_2H + k_3I + k_4J + k_5K \\ & + k_6L + k_7M + k_8N + k_9O + k_{10}P \\ & + k_{11}Q + k_{12}R + k_{13}S + k_{14}T \end{aligned} \tag{23}$$

$$TT = F \tag{24}$$

C. QUANTUM CIRCUIT FOR TOOM-COOK 8-WAY MULTIPLICATION

Figure 5 −8 shows a high-level quantum circuit for constructing the Toom-Cook 8-way multiplication. The quantum circuit generally follows the computational steps described previously, e.g., the splitting process, evaluation points, pointwise multiplication, interpolation, and recomposition. The quantum Toom-Cook 8-way multiplication necessitates five underlying operations, which include copy, addition, subtraction, shift, and the underlying sub multiplication operations.

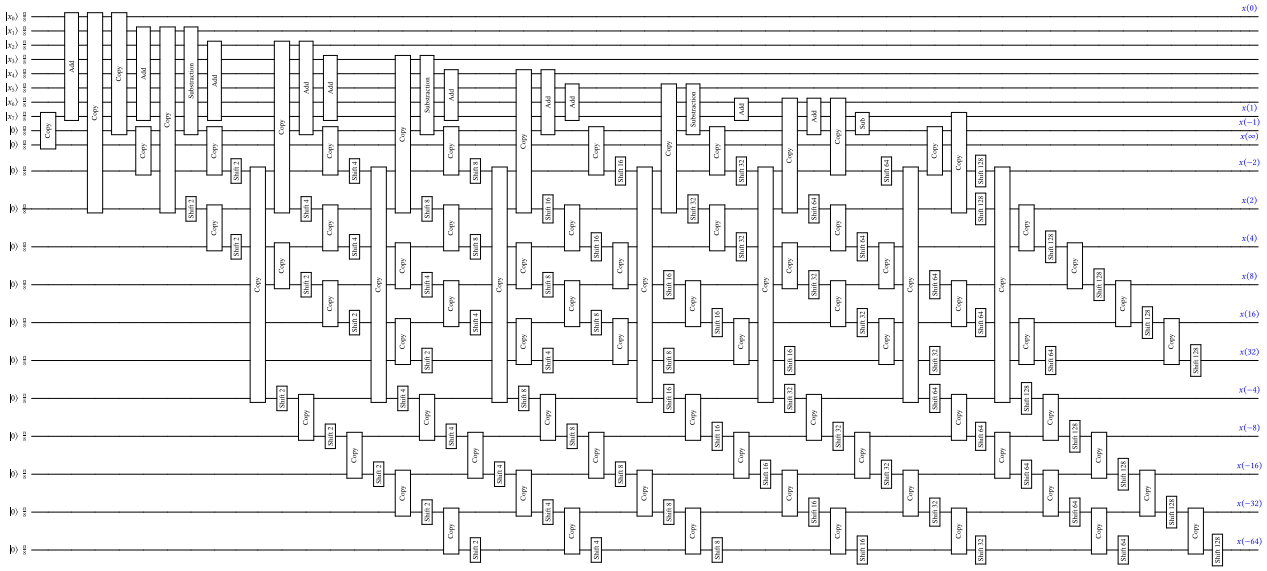

FIGURE 6. Quantum Circuit of evaluation point x.

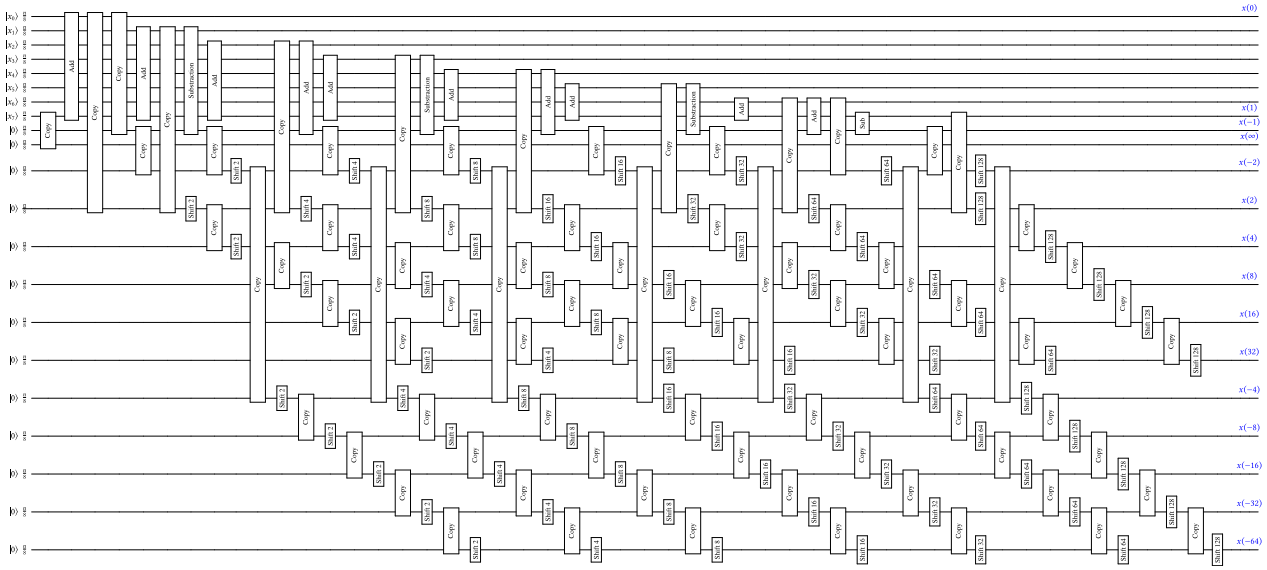

FIGURE 7. Quantum Circuit of evaluation point y.

In order to accomplish the input splitting step, we first place the $n-$bit number x as the initial multiplicand in each of the registers designated x_0, x_1, x_2, x_3, x_4, x_5, x_6, and x_7. This separates x into eight pieces, each of which is $\frac{n}{8}$ in length. Accordingly, the evaluation of each x and y is depicted in Figures 6 and 7, which all run simultaneously due to the fact that they operate on separate registers. Figure 8 depicts the point-wise multiplication and interpolation performed by the Toom-Cook 8-way quantum circuit.

Each result of interpolation has an $\frac{2n}{8}$-bit size equal to TT, SS, RR, QQ, PP, OO, NN, MM, LL, KK, JJ, II, HH, GG, and FF. In the last stage, recomposition combines the interpolation results that initially required 190 $\frac{n}{4}$-bit addition circuit operations, and without overlapping the adds.

V. RESULTS AND ANALYSIS

In this section, we will discuss the results of the Toom-Cook 8-way circuit design and their complexity, as shown in Table 1, by the metrics of cost multiplication to the Toffoli count, qubit count, and Toffoli depth. For the analysis of gate-count calculations on multipliers, we use the same basic scenario assumption to calculate the gate count as [2], [7], and [8], that one Toffoli gate can be used to perform one-bit number multiplication, and the cost of the in-place adder A_n must be less than or equal to $2n$ Toffoli gates, where n is the bit size of the larger addend, in order to compare the final result equivalently to the previous study.

In the Larasati et al. study, they use the division circuit so that they can consider strategies to implement the constant

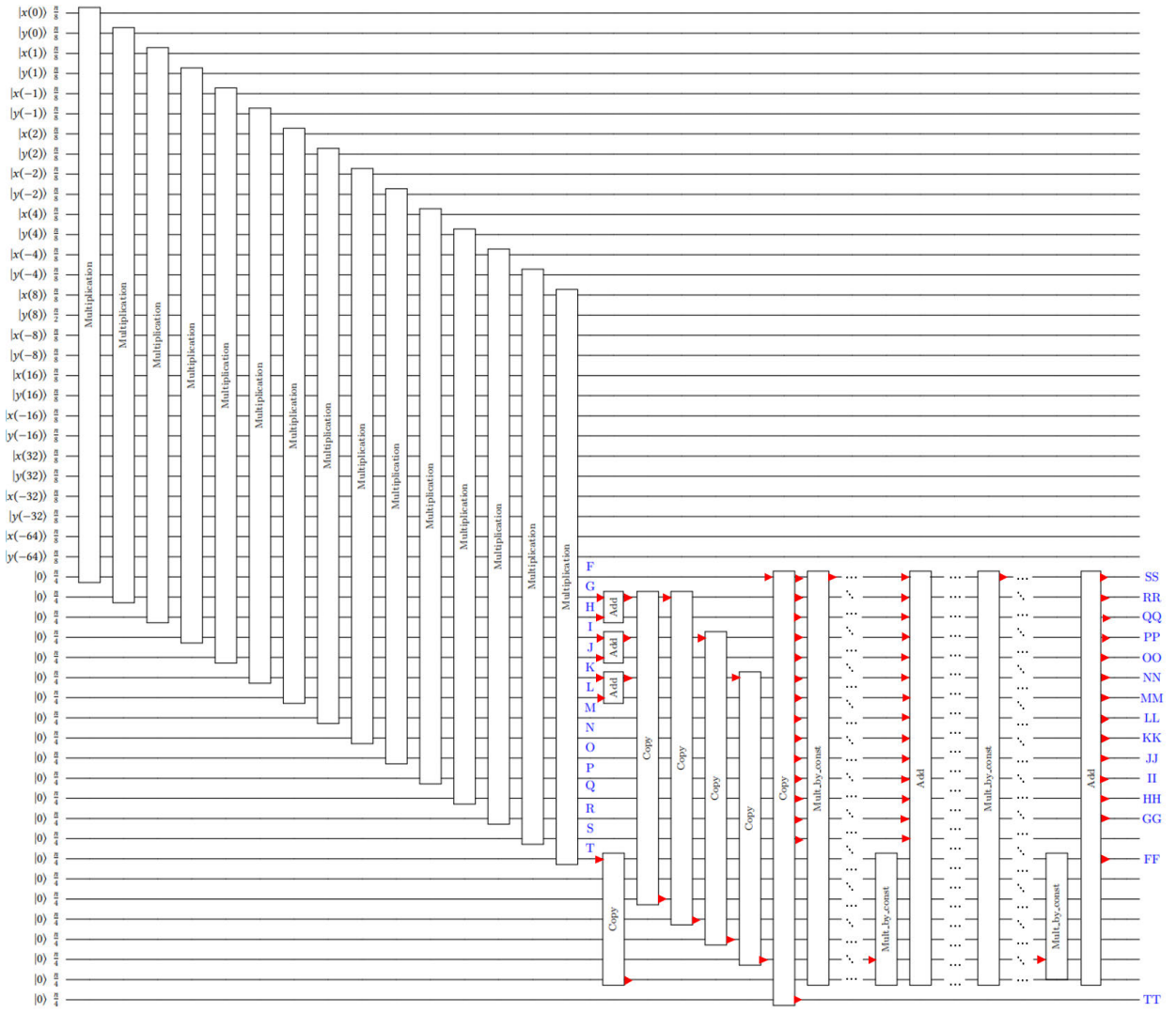

FIGURE 8. Quantum circuit consisting of point-wise multiplication and the interpolation of the Toom-Cook 8-way. The image depicts, in the form of function block boxes, the six different processes that, when combined, make up the Toom-Cook 8-way quantum circuit. The operations consist of copying, subtracting, adding, shifting, and sub-multiplying the data.

multiplication from [36] by performing a Toffoli count of $4n(n+1)$ to achieve the lowest feasible cost while conducting quantum division. In contrast to the Toom Cook 3-way design by Larasati et al., this study does not use division to simplify the evaluation of points x and y, thereby reducing costs even further. This will impact the overall efficiency and depth of quantum multiplication. In the Toom Cook 8-way design, despite using the division function, we only use the copy and add functions, which only require the CNOT gate. Thus, the space and time efficiency of this Toom-Cook 8-way can be improved.

A. TOFFOLI GATE COUNT

Let T_n represent the cost of using the Toom-Cook 8-way multiplier to multiply two larger n-bit values. As a result, A_n represents the cost of adding or subtracting n-bits. To realize n-bit Toom-Cook 8-way multiplication, 15 operations $\frac{n}{8}$ submultiplications and three adder types of varying lengths (28 operations $\frac{n}{8}$-bit, 190 operations $\frac{2n}{8}$-bit, and five operations $\frac{n}{8}$-bit addition and subtraction) are required. Equation 25 is then used to determine the Toffoli cost of an n-bit Toom-Cook 8-way multiplication. Furthermore, for recursive implementations, the cost rises to Equation 26, and Equation 27 becomes equivalent when the Toffoli cost of $A_n = 2n$ is substituted.

$$T_n = 15T_{\frac{n}{8}} + 28A_{\frac{n}{8}} + 190A_{\frac{n}{4}} + 5\,A_{\frac{3n}{8}} \tag{25}$$

$$\begin{aligned} T_n = {} & 15^{\log_8 n}T_1 + 28(A\frac{n}{8} + 14A\frac{n}{64} + \cdots + 14^{\log_8(n)-1}A_1) \\ & + 190(A\frac{n}{4} + 95A\frac{n}{32} + \cdots + 95^{\log_8(n)-1}A_2) \\ & + 5(A\frac{3n}{8} + 2\,A\frac{3n}{64} + \cdots + 2^{\log_8(n)-1}A_3) \end{aligned} \tag{26}$$

$$T_n = 15^{\log_8 n} + \sum_{i=0}^{\log_8(n)-1} \left[56n(\frac{15}{8})^i\right] \tag{27}$$

Using the geometric series $\sum_{i=0}^{m-1} r^i = \frac{1-r^m}{1-r}$, we can calculate the Toffoli cost of recursive implementation, denoted by Equation 28.

$$\begin{aligned} T_n &= 15^{\log_8 n} + 56n\left(\frac{1-(\frac{15}{8})^{\log_8 n}}{1-(\frac{15}{8})}\right) \\ &= n^{\log_8 15} + 56n\left(\frac{1-n^{\log_8(\frac{15}{8})}}{1-(\frac{15}{8})}\right) \\ &= 57n^{\log_8 15} - 64n \end{aligned} \tag{28}$$

The obtained result of Equation 28 does not account for the typical uncomputation procedure performed in a quantum environment. This strategy is also considered in [2], [7], and [8] research, so Equation 29 incorporates the uncomputed process to avoid roughly doubling the previously acquired cost. Note that we employ the same definition of ''clean cost'' as Larasati et al. in the following equation.

$$T_{n(clean)} = 112n^{\log_8 15} - 128n \tag{29}$$

B. SPACE-TIME COMPLEXITY ANALYSIS

Bennett in [41] introduced the technique of reversible pebble games for measuring asymptotic performance improvements in the context of space consumption in the context of space-time complexity analysis. This technique is utilized extensively in reversible computing, which makes time and space complexity analysis possible and enables time-efficient finite-space computing [42]. This method will allow us to evaluate the difference in the cost of the successfully optimized multiplication and compare it to the results of previous studies. We determined the optimal cost of multiplication by following the procedures outlined in [2], [7], and [8].

In the Toom-Cook 8-way algorithm, 15 simultaneous multiplications were done in a recursive way to make a quinary eight structure. There are 15^l nodes of size $3^{-l}n$ for an input of size n at level l, and this input has a total circuit cost of $n(\frac{15}{8})^l$. Equation 30depicts the total price of the quinary tree. For determining the optimal tree height k for optimal performance, use Equation 32.

$$n\sum_{i=0}^{N}\left(\frac{15}{8}\right)^i, \quad N = \log_8 n \tag{30}$$

$$n\sum_{i=0}^{N-k-1}\left(\frac{15}{8}\right)^i = \frac{1}{8^{N-k}}\sum_{i=0}^{k-1}\left(\frac{15}{8}\right)^i \tag{31}$$

In a pattern similar to Equation 29, the identity of the geometric series enables us to locate the boundaries indicated by Equation 32. Thus, the space can be reduced, as shown in qubit count Equation 33. The obtained result from Equation 33, approximately equal to $\mathcal{O}(n^{1.245})$, is lower than the initially required space assessed with Equation 34, which is confined to the value $\mathcal{O}(n^{\log_8 15}) \approx \mathcal{O}(n^{1.30229})$.

$$k \leq \frac{N}{2-\frac{\log 8}{\log 15}} \approx 0.8116 \tag{32}$$

$$\begin{aligned} QC &= \mathcal{O}\left(n\left(\frac{15}{8}^{\left(\frac{\log 15}{2\log 15-2\log 8}\right)\log_8 n}\right)\right) \\ &\approx \mathcal{O}(n^{1.245}) \end{aligned} \tag{33}$$

$$n\sum_{k=0}^{\log_8 n-1}\left(\frac{15}{8}\right)^k = n\left(\frac{1-\left(\frac{15}{8}\right)^{\log_8 n}}{1-\frac{15}{8}}\right) \tag{34}$$

The Toffoli depth of a circuit is a prevalent way to describe its time complexity [7], [43]. It can be calculated by multiplying the number of subtrees S_k at the $k-th$ level by the corresponding depth D_k. Consequently, we can express the Toffoli depth T_d as in Equation 35.

$$\begin{aligned} S_k &= 15^{\left(1-\frac{\log 15}{2\log 15-log8}\right)\log_8 n} \\ D_k &= \frac{n}{8^{\left(1-\frac{\log 15}{2\log 15-log8}\right)\log_8 n}} \\ T_d &= S_k D_k = n\left(\frac{15}{8}\right)^{\left(1-\frac{\log 15}{2\log 15-log8}\right)\log_8 n} \approx n^{1.0569} \end{aligned} \tag{35}$$

C. COMPLEXITY ANALYSIS COMPARISON

The schoolbook multiplication, or a naïve approach similar to Toom-Cook 1-way, requires $\mathcal{O}(n^2)$ steps, which is quadratic in the amount of the input; this is also the value for Naive's Toffoli depth. In addition, an enhanced study in 2004 places Naive's Toffoli depth at $\mathcal{O}(n \log n)$ [44]. For asymptotic performance analysis in the quantum implementation, the schoolbook method requires $\mathcal{O}(n)$ for the qubit count and $\mathcal{O}(n^2)$ for the Toffoli count and depth values. Quantum multiplication costs involved are $(4n+1)$ qubit count, $(4n^2-4n+1)$ Toffoli depth, and $(4n^2-3n)$ Toffoli count [7], [8]. Afterward, some studies in multiplication, i.e., Toom-Cook 2-way (Karatsuba), Toom-Cook 3-way, etc., attempt to obtain the minor step or complexity.

Asymptotic performance study in Karatsuba multiplication yielded qubit count $\mathcal{O}(n^{\log_2(3)})$ for qubit count and Toffoli count, as well as the same result value for CNOT consumption [1], [3] [15]. Improvements to the value of asymptotic performance analysis and cost for quantum implementation for the multiplication of Karatsuba can be referred from Parent et al.'s research, where the asymptotic values for qubit count $\mathcal{O}(n^{1.427})$, the same $\mathcal{O}(n^{\log_2(3)})$ for Toffoli count, and Toffoli depth $\mathcal{O}(n^{1.158})$ [2], [7] [8].

In Parent et al. study, the value of the qubit count $n^{1.427}$, the Toffoli count $\mathcal{O}(n^{\log_2 3})$, and the Toffoli depth $n^{1.158}$, which then for the qubit count value of asymptotic performance analysis the quantum Karatsuba multiplier implementation, were improved by van Hoof's research in [3] with value $3n$ for the qubit count. The Karatsuba variant from Putranto et al. is reported to have the lowest CNOT usage from $\mathcal{O}(n^2)$ CNOT in van Hoof to $\mathcal{O}(n^{\log_2(3)})$ which claimed as compliment research from previous, a time-efficient based while preserving space-efficient quantum multiplication implementation by van Hoof [15] and Banegas et al. [40].

TABLE 1. Comparison of multipliers' asymptotic performance and implementation cost using the toffoli count, qubit count, and toffoli depth as space-time complexity metrics. to give a thorough comparison of the development of complexity multiplication research for the naive schoolbook, karatsuba, and toom-cook-based studies from prior studies, we present our findings for toom-cook 2-way, 4-way, and 8-way.

No	Reference	Arithmetic Algorithm	Asymptotic Performance Analysis			Cost of Quantum Implementation of Multiplication			
			Qubit Count	Toffoli Count	Toffoli Depth	Qubit Count	Toffoli Count	Toffoli Depth	CNOT
1	Naïve (2004, [2])	schoolbook	$O(n)$	$O(n^2)$	$O(n^2)$	$4n+1$	$4n^2+3n$	$4n^2-4n+1$	-
2	Naïve improved (2004, [44])	schoolbook	$O(n)$	$O(n^2)$	$O(n\log n)$	-	-	-	-
3	Paldivis et al (2014, [36])	Const Mult	$O(n)$	$O(n^2)$	$O(n)$	$3n+1$	$4n(n+1)$	8n	-
4	Kepley and Steinwandt (2015, [1])	Karatsuba	$O(n^{\log_2 3})$	$O(n^{\log_2^3})$	-	-	-	-	$O(n^{\log_2 3})$
5	Parent et.al. (2017, [2])	Karatsuba	$O(n^{1.427})$	$O(n^{\log_2 3})$	$O(n^{1.158})$	$n(\frac{3}{2})^{\frac{\log 2}{(2\log 3-\log 2)}\log_2 n}\approx n^{1.427}$	$42n^{\log_2 3}$	$n(\frac{3}{2})^{1-\frac{\log 3}{(2\log 3-\log 2)}\log_2 n}\approx n^{1.158}$	-
6	Dutta et al (2018, [7])	Toom-Cook-2.5	$O(n^{1.404})$	$O(n^{\log_6^{16}})$	$O(n^{1.143})$	$n(\frac{8}{3})^{\frac{\log 16}{(6\log 16-\log 6)}\log_6 n}\approx n^{1.404}$	$49n^{log_6 16}$	$n(\frac{8}{3})^{1-\frac{\log 16}{(2\log 16-\log 6)}\log_6 n}\approx n^{1.143}$	-
8	Larasati(2021, [8])	Toom-Cook 3	$O(n^{1.35})$	$O(n^2)$	$O(n^{1.112})$	$n(\frac{5}{3})^{\frac{\log 5}{(2\log 5-\log 3)}\log_3 n}\approx n^{1.353}$	$8n^2+66n^{\log_3 5}-72$	$n(\frac{5}{3})^{1-\frac{\log 5}{(2\log 5-\log 3)}\log_3 n}\approx n^{1.112}$	-
9	Van Hoof (2020, [3])	Karatsuba	$3n$	$O(n^{log_2 3})$	-	-	-	-	$O(n^2)$
10	Putranto et al (2022, [15])	Karatsuba	$3n$	$O(n^{log_2 3})$	-	-	-	-	$O(n^{log_2 3})$
11	Ours	Toom Cook 2-way	$O(n^{1.589})$	$O(n^{\log_2 3})$	$O(n^{1.217})$	$n(\frac{3}{2})^{\frac{\log 3}{(2\log 3-\log 2)}\log_2 n}\approx n^{1.589}$	$34n^{\log_2 3}-32n$	$n(\frac{3}{2})^{1-\frac{\log 3}{(2\log 3-\log 2)}\log_2 n}\approx n^{1.217}$	-
12	Ours	Toom Cook 4-way	$O(n^{1.313})$	$O(n^{\log_4 7})$	$O(n^{1.09})$	$n(\frac{7}{4})^{\frac{\log 7}{(2\log 7-\log 4)}\log_4 n}\approx n^{1.313}$	$122n^{\log_4 7}-160n$	$n(\frac{7}{4})^{1-\frac{\log 7}{(2\log 7-\log 4)}\log_4 n}\approx n^{1.09}$	-
13	Ours	Toom Cook 8-way	$O(n^{1.245})$	$O(n^{\log_8 15})$	$O(n^{1.0569})$	$n(\frac{15}{8})^{\frac{\log 15}{(2\log 15-\log 8)}\log_8 n}\approx n^{1.245}$	$112n^{\log_8 15}-128n$	$n(\frac{15}{8})^{1-\frac{\log 15}{(2\log 15-\log 8)}\log_8 n}\approx n^{1.0569}$	-

Paldivis's research on the complexity of the Toom-Cook-based multiplication obtained values of $\mathcal{O}(n)$ Qubit count, $\mathcal{O}(n^2)$ Toffoli Count, and $\mathcal{O}(n)$ Toffoli Depth. The multiplication implementation to design Shor's quantum factorization requires $3n+1$ qubits, $4n(n+1)$ Toffoli, and $8n$ Toffoli depths [36]. Dutta et al. study [7], which Larasati et.al. also referred to in [8], improved the asymptotic performance analysis values for the qubit count of $n^{1.404}$, Toffoli count $\mathcal{O}(n\log_6 16)$, and Toffoli depth $n^{1.143}$. With the values of qubit count $n^{1.404}$, Toffoli count $49n^{\log_6 16}$, and Toffoli depth $n^{1.143}$, the cost of implementing quantum is reduced [7].

An estimate of the asymptotic performance analysis value of $n^{1.353}$ qubit count, $\mathcal{O}(n^2)$ Toffoli count, and $n^{1.112}$ for Toffoli depth is given by recent research on Toom-Cook multiplication by Larasati et al. As the research that also underlies this research, Larasati et al.'s research is intended to be the lowest in implementing the cost quantum multiplier, i.e., the number of qubits, Toffoli, and depth.

As shown in the Table 1, Toom-Cook 8-way quantum multiplier has asymptotic performance analysis in qubit count with $n(\frac{15}{8})^{\frac{\log 15}{(2\log 15-\log 8)}\log_8 n}$ approximately $\mathcal{O}(n^{1.245})$. In terms of Toffoli depth, related to fast computation, the design yields a lower logical depth bound to $\mathcal{O}(n^{1.0569})$ and Toffoli count $\mathcal{O}(n^{\log_8 15})$. Compared to the other multiplication, the results show that the designed Toom-Cook 8-way multiplier has the lowest asymptotic performance for qubit count, Toffoli count, and Toffoli depth, consequently resulting in more sub-multiplication iterations than previous work.

In particular, in terms of cost of quantum implementation, the Toom-Cook 8-way with Toffoli count $112n^{\log_8 15}-128n$ cost scale is smaller than the 3-way Toom-Cook with $8n^2+66n^{\log_3 5}-72$, and efficiency is much better than other Toom-Cook-based multiplier, Karatsuba, and Naive Schoolbooks. Similarly, the number of qubit count $n(\frac{15}{8})^{\frac{\log 15}{(2\log 15-\log 8)}\log_8 n}\approx n^{1.245}$ for Toom-Cook 8-way also outperforms the efficiency of other Toom-Cook-based multiplier, especially Toom-Cook 4-way, which is currently used in lattice-based algorithms.

Comparing our $\mathcal{O}(n^{1.0569})$ to other prior research, we note that in Palvidis's research, Toffoli depth reported $\mathcal{O}(n)$ in their constant multiplication ConstMult Multiplier, but they may have a more significant cost implementation. Also, in the research from Larasati et al., in the case of Toffoli's count, the Toom-Cook 3-way series with $\mathcal{O}(n)$ is still on a quadratic scale, similar to the naïve technique, with constant multiplication or division operations contributing significantly to its implementation [8]. The result for the Toffoli depth value is $112n^{\log_8 15}-128n$, which has proven more efficiency reached and has the lowest cost.

VI. DISCUSSION

With a Toffoli depth of $O(n^{1.0569})$ and a qubit count of $O(n^{1.245})$ in Toom-Cook 8-way, the designed multiplier also serves as an appropriate follow-up to prove the statement from Larasati et al.'s research [8] that a higher order in Toom-k multiplication will give higher efficiency. Despite the fact that numerous studies on classical calculation concur that the Toom-Cook algorithm is fast, the [45] study asserts that multiplication is likely inaccurate. The Toom-Cook algorithm for multiplying polynomials may encounter difficulties due to an interpolation step that requires dividing by an even number. Although we also discover that this inaccuracy is a result of the fractional results of the inverse matrix in the interpolation, this issue can be resolved by changing the variable type in the multiplication algorithm to float. Nevertheless, as indicated by the [6] study, which does not use non-trivial division, we also employ a balanced 8-way Toom-Cook operation without division strategy to reduce this inaccuracy issue. Inaccuracy may affect the use of cryptography, but in our research, it does not have much effect, considering that the asymptotic digit values obtained are close to the actual value. However, this paper has not discussed several challenges, including Toom-Cook multiplication accuracy at a higher power of 8 or process acceleration in hardware implementations such as FPGAs, as in [20]. At a higher order, inaccuracy

may occur, according to [45]'s research. Furthermore, this research only limits the applied quantum circuit to 8 ways so that it can be reliable on quantum processor hardware, which has only reached a 53-qubit quantum state.

According to the findings of Mujdei et al. [10], the Toom-Cook is implemented in the lattice-based PQC algorithm, and it can be attacked in a straightforward manner. Toom Cook with higher sub-operations, in this case, Toom-Cook 8-way, can generate more sub-operation iterations of schoolbook multiplication, potentially resulting in a lower linear threshold or a decline in peak, as shown by the variance plot findings in the CPA-based attack research. As a result, using space- and time-efficient multiplication, such as Toom-Cook 8-way, is critical not only for making the best use of resources with asymptotic performance or for implementing at the lowest cost, but also for avoiding attacks like CPA-based attacks. In this study, we provide a brief overview of attacks resulting from the exploitation of elementary arithmetic, such as correlation power analysis. However, neither the SCA attack nor the combined SCA attacks were described in detail. In addition, the designed implementation of multiplication for the PQC algorithm is not explained. Future research would benefit from including more details on how to attack or measure its effects, as this would result in a more technically extensive analysis of the multiplication-based attack and its implementation in a lattice-based
PQC algorithm.

VII. CONCLUSION

This research examines several prior studies on multiplication complexity in classical and quantum environment, including the Naive schoolbook, Karatsuba, and Toom-Cook-based multiplication. We provide insight into the function of multiplication in the PQC era, including PQC algorithms based on Toom-Cook multiplication, multiplication as an exploitation object in CPA-based SCA attacks, and the need for efficient arithmetic to design a quantum cryptanalysis circuit. The existing polynomial multiplication based on Toom-Cook is straightforward to attack, as proved by prior research with the Toom-Cook 4-way SCA attack. We created a multiplier with more schoolbook multiplier sub-operations and a quantum multiplier that uses less space and time to change the number of peaks in the CPA analysis. The Toom-Cook 8-way quantum multiplier has the lowest asymptotic performance analysis when compared to the Naive schoolbook, Karatsuba, and existent Toom-Cook-based multipliers. The Toom-Cook 8-way multiplier employs numerous strategies from earlier research, such as not using the division gate. It achieves the best asymptotic performance analysis and cost implementation to impact space and time efficiency. This yields a qubit count of $n(\frac{15}{8})^{\frac{\log 15}{(2\log 15-\log 8)}\log_8 n}$, or about $\mathcal{O}(n^{1.245})$, and Toffoli counts and depths of $112n^{\log_8 15} - 128n$ and $n(\frac{15}{8})^{1-\frac{\log 15}{(2\log 15-\log 8)}\log_8 n} \approx n^{1.0569}$, respectively. The increased likelihood of sub-multiplication iterations, which will impact the analysis of SCA based on CPA, results from the multiplication design's greater efficiency. Thus, using a multiplier in the PQC era depends on speed, efficiency, and the prevention of multiplication-based attacks. In the classical implementation, this multiplier design is expected to be a constructor for a post-quantum algorithm that is mostly based on lattices.

REFERENCES

[1] S. Kepley and R. Steinwandt, "Quantum circuits for $\mathbb{F}_{2^n}$-multiplication with subquadratic gate count," *Quantum Inf. Process.*, vol. 14, no. 7, pp. 2373–2386, 2015.

[2] A. Parent, M. Roetteler, and M. Mosca, "Improved reversible and quantum circuits for Karatsuba-based integer multiplication," in *Proc. 12th Conf. Theory Quantum Comput., Commun., Cryptogr.* Cham, Switzerland: Springer, 2017, pp. 7:1–7:15.

[3] I. van Hoof, "Space-efficient quantum multiplication polynomials for binary finite fields with sub-quadratic Toffoli gate count," *Quantum Inf. Comput.*, vol. 20, no. 9, pp. 721–735, 2020, doi: 10.26421/QIC20.9-10-1.

[4] P. L. Montgomery, "Five, six, and seven-term Karatsuba-like formulae," *IEEE Trans. Comput.*, vol. 54, no. 3, pp. 362–369, Mar. 2005.

[5] A. L. Toom, "The complexity of a scheme of functional elements realizing the multiplication of integers," *Soviet Math. Doklady*, vol. 3, no. 4, pp. 714–716, 1963.

[6] A. Zanoni, "Toom-Cook 8-way for long integers multiplication," in *Proc. 11th Int. Symp. Symbolic Numeric Algorithms Sci. Comput.*, Sep. 2009, pp. 54–57.

[7] S. Dutta, D. Bhattacharjee, and A. Chattopadhyay, "Quantum circuits for Toom-Cook multiplication," *Phys. Rev. A, Gen. Phys.*, vol. 98, no. 1, Jul. 2018, Art. no. 012311.

[8] H. T. Larasati, A. M. Awaludin, J. Ji, and H. Kim, "Quantum circuit design of Toom 3-way multiplication," *Appl. Sci.*, vol. 11, no. 9, p. 3752, Apr. 2021.

[9] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, and J. Lichtinger, "Status report on the third round of the NIST post-quantum cryptography standardization process," NIST, U.S. Dept. Commerce, Washington, DC, USA, Tech. Rep. 8413, 2022.

[10] C. Mujdei, L. Wouters, A. Karmakar, A. Beckers, J. M. B. Mera, and I. Verbauwhede, "Side-channel analysis of lattice-based post-quantum cryptography: Exploiting polynomial multiplication," *ACM Trans. Embedded Comput. Syst.*, vol. 2022, pp. 1–26, Nov. 2022.

[11] A. Karatsuba and Y. Ofman, "Multiplication of many-digital numbers by automatic computers," *Doklady Akademii Nauk SSSR*, vol. 145, no. 2, pp. 293–294, 1962.

[12] D. Maslov, J. Mathew, D. Cheung, and D. K. Pradhan, "An $O(m^2)$-depth quantum algorithm for the elliptic curve discrete logarithm problem over $GF(2^m)^a$," *Quantum Inf. Comput.*, vol. 9, no. 7, pp. 610–621, Jul. 2009.

[13] S. A. Cook and S. O. Aanderaa, "On the minimum computation time of functions," *Trans. Amer. Math. Soc.*, vol. 142, pp. 291–314, Aug. 1969.

[14] M. Bodrato, "Towards optimal Toom-Cook multiplication for univariate and multivariate polynomials in characteristic 2 and 0," in *Proc. Int. Workshop Arithmetic Finite Fields*. Cham, Switzerland: Springer, 2007, pp. 116–133.

[15] D. S. C. Putranto, R. W. Wardhani, H. T. Larasati, and H. Kim, "Another concrete quantum cryptanalysis of binary elliptic curves," Cryptol. ePrint Arch., Paper 2022/501, 2022. [Online]. Available: https://eprint.iacr.org/2022/501

[16] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle, "CRYSTALS—Kyber: A CCA-secure module-lattice-based KEM," in *Proc. IEEE Eur. Symp. Secur. Privacy*, Apr. 2018, pp. 353–367.

[17] V. Lyubashevsky, "Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Cham, Switzerland: Springer, 2009, pp. 598–616.

[18] D. Micciancio and O. Regev, "Post-quantum cryptography, chapter lattice-based cryptography," *Computing*, vol. 85, nos. 1–2, pp. 105–125, 2008.

[19] Z. Liu, K.-K. R. Choo, and J. Großschädl, "Securing edge devices in the post-quantum Internet of Things using lattice-based cryptography," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 158–162, Feb. 2018.

[20] M. Bisheh-Niasar, R. Azarderakhsh, and M. Mozaffari-Kermani, "Instruction-set accelerated implementation of CRYSTALS-Kyber," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 11, pp. 4648–4659, Nov. 2021.
[21] M. Bisheh-Niasar, R. Azarderakhsh, and M. Mozaffari-Kermani, "High-speed NTT-based polynomial multiplication accelerator for post-quantum cryptography," in *Proc. IEEE 28th Symp. Comput. Arithmetic (ARITH)*, Jun. 2021, pp. 94–101.
[22] J. M. Pollard, "The fast Fourier transform in a finite field," *Math. Comput.*, vol. 25, no. 114, pp. 365–374, Apr. 1971.
[23] H. Pahlevanzadeh, J. Dofe, and Q. Yu, "Assessing CPA resistance of AES with different fault tolerance mechanisms," in *Proc. 21st Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2016, pp. 661–666.
[24] T. Schneider, A. Moradi, and T. Güneysu, "ParTI—Towards combined hardware countermeasures against side-channel and fault-injection attacks," in *Proc. Annu. Int. Cryptol. Conf.* Cham, Switzerland: Springer, 2016, pp. 302–332.
[25] F. Regazzoni, T. Eisenbarth, L. Breveglieri, P. Ienne, and I. Koren, "Can knowledge regarding the presence of countermeasures against fault attacks simplify power attacks on cryptographic devices?" in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, Oct. 2008, pp. 202–210.
[26] F. Regazzoni, T. Eisenbarth, J. Grobschadl, L. Breveglieri, P. Ienne, I. Koren, and C. Paar, "Power attacks resistance of cryptographic S-boxes with added error detection circuits," in *Proc. 22nd IEEE Int. Symp. Defect Fault-Tolerance VLSI Syst.*, Sep. 2007, pp. 508–516.
[27] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Reliable hardware architectures for the third-round SHA-3 finalist Grostl benchmarked on FPGA platform," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst.*, Oct. 2011, pp. 325–331.
[28] J. Dofe, H. Pahlevanzadeh, and Q. Yu, "A comprehensive FPGA-based assessment on fault-resistant AES against correlation power analysis attack," *J. Electron. Test.*, vol. 32, no. 5, pp. 611–624, Oct. 2016.
[29] C. Whelan and M. Scott, "Side channel analysis of practical pairing implementations: Which path is more secure?" in *Proc. Int. Conf. Cryptol. Vietnam*. Cham, Switzerland: Springer, 2006, pp. 99–114.
[30] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Annu. Int. Cryptol. Conf.* Cham, Switzerland: Springer, 1999, pp. 388–397.
[31] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2004, pp. 16–29.
[32] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 124–134.
[33] V. Vedral, A. Barenco, and A. Ekert, "Quantum networks for elementary arithmetic operations," *Phys. Rev. A, Gen. Phys.*, vol. 54, p. 147, Jul. 1996.
[34] D. Beckman, A. N. Chari, S. Devabhaktuni, and J. Preskill, "Efficient networks for quantum factoring," *Phys. Rev. A, Gen. Phys.*, vol. 54, no. 2, pp. 1034–1063, Aug. 1996.
[35] S. Beauregard, "Circuit for Shor's algorithm using 2n+3 qubits," 2002, *arXiv:quant-ph/0205095*.
[36] A. Pavlidis and D. Gizopoulos, "Fast quantum modular exponentiation architecture for Shor's factoring alogrithm," *Quantum Inf. Comput.*, vol. 14, no. 8, pp. 649–682, May 2014.
[37] R. Rines and I. Chuang, "High performance quantum modular multipliers," 2018, *arXiv:1801.01081*.
[38] D. S. Roche, "Space- and time-efficient polynomial multiplication," in *Proc. Int. Symp. Symbolic Algebr. Comput.*, Jul. 2009, pp. 295–302.
[39] C. Gidney, "Asymptotically efficient quantum Karatsuba multiplication," 2019, *arXiv:1904.07356*.
[40] G. Banegas, D. J. Bernstein, I. Van Hoof, and T. Lange, "Concrete quantum cryptanalysis of binary elliptic curves," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2020, pp. 451–472, Dec. 2020.
[41] C. H. Bennett, "Time/space trade-offs for reversible computation," *SIAM J. Comput.*, vol. 18, no. 4, pp. 766–776, Aug. 1989.
[42] R. Královič, "Time and space complexity of reversible pebbling," in *Proc. Int. Conf. Current Trends Theory Pract. Comput. Sci.* Cham, Switzerland: Springer, 2001, pp. 292–303.
[43] M. Amy, "Algorithms for the optimization of quantum circuits," M.S. thesis, Dept. Comput. Sci., Quantum Inf., Univ. Waterloo, Waterloo, ON, Canada, 2013.
[44] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, "A logarithmic-depth quantum carry-lookahead adder," *Quantum Inf. Comput.*, vol. 6, no. 5, pp. 351–369, Jul. 2006.
[45] M. Elia, *Loss of Precision in Implementations of the Toom-Cook Algorithm*. Burlington, VT, USA: The Univ. Vermont State Agricult. College, 2021.

DEDY SEPTONO CATUR PUTRANTO (Member, IEEE) received the Dr.Eng. degree in nano integrated systems from Shizuoka University, Hamamatsu, Japan, in 2015. He worked with the National Cyber and Crypto Agency, Indonesia, from 2003 to 2022. He is currently a Postdoctoral Researcher with the Department of Computer Science and Engineering, Pusan National University, South Korea. His research interests include hardware security, information security, cryptography, and quantum computing.

RINI WISNU WARDHANI (Graduate Student Member, IEEE) received the M.Eng. degree in electrical engineering from the University of Indonesia, in 2011. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Pusan National University, South Korea. She was with National Cyber and Crypto Agency, Indonesia, from 2003 to 2021. Her research interests include hardware security, information security, cryptography, and quantum computing.

HARASHTA TATIMMA LARASATI (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in telecommunication engineering from the Institut Teknologi Bandung (ITB), Bandung, Indonesia, in 2016 and 2017, respectively. She is currently pursuing the Ph.D. degree in computer engineering with Pusan National University, Busan, South Korea. She is currently a Junior Lecturer with her home university, ITB. Her research interests include quantum computing and cryptanalysis, quantum machine learning, AI security, and networking.

HOWON KIM (Member, IEEE) received the bachelor's degree from Kyungpook National University (KNU) and the Ph.D. degree from the Pohang University of Science and Technology (POSTECH). He is currently a Professor with the Department of Computer Science and Engineering, the Chief of Energy Internet of Things (IoT), IT Research Center (ITRC), and the Chief of Information Security Education Center (ISEC), Pusan National University (PNU). Before joining PNU, he worked with the Electronics and Telecommunications Research Institute (ETRI), as a Team Leader, for ten years beginning from December 1998. He was a Visiting Postdoctoral Researcher with the Communication Security Group (COSY), Ruhr-Universität Bochum, Germany, from July 2003 to June 2004.