High-efficiency Low-latency NTT Polynomial Multiplier for Ring-LWE Cryptography

Tuy Nguyen Tan¹, Tram Thi Bao Nguyen², and Hanho Lee²∗

Abstract—This paper presents a novel architecture to perform polynomial multiplication in ring learning with errors (ring-LWE) cryptosystems. By employing number theoretic transform (NTT) of the input polynomials simultaneously, the multiplication latency is significantly reduced. In addition, a multiple-path delay feedback (MDF) architecture is used to speed up the multiplication process. As a result, the proposed NTT multiplier offers a better value of area-latency product compared with that of previous studies. The simulation results for the security parameters \( n = 512 \) and \( q = 12,289 \) on Xilinx Virtex-7 FPGA show that the proposed multiplier uses only about 8.69% of the number of clock cycles required by previous works to completely perform the polynomial multiplication. Furthermore, the obtained area-latency product value of the proposed architecture is less than 45.3% of that of previous works.

Index Terms—Cryptography, multiple-path delay feedback, NTT polynomial multiplier, ring-LWE

I. INTRODUCTION

Conventional cryptosystems such as RSA and elliptic curve cryptography (ECC) can be solved in polynomial time with the algorithms proposed by Shor using a quantum computer [1]. To replace these classic cryptosystems, post-quantum cryptosystems offering higher levels of security have been researched recently. Among the existing post-quantum cryptosystems, ring-LWE cryptography is a promising candidate because its security is based on the worst-case hardness problem [2]. The basic concept of ring-LWE cryptography can be found in [2, 3]. The block diagram of a ring-LWE cryptosystem is shown in Fig. 1. The input message \( m \) is encrypted into ciphertext \((c_1, c_2)\) using arithmetic computation on public key \((a, p)\) and error polynomials \(e_1, e_2, \) and \(e_3\). Original message \( m \) can be recovered from ciphertext \((c_1, c_2)\) and private key \(r_2\) through the decryption operation. The arithmetic operations in ring-LWE cryptography include polynomial addition, modulus, and polynomial multiplication, in which polynomial multiplication is the basic and most computationally intensive operation [4].

Several architectures to perform polynomial multiplication have been introduced in literatures. Among the existing approaches used to conduct polynomial multiplication, the number theoretic transform (NTT) studied in [4-6] is an efficient method. Let \( a(x) \) be a polynomial over the ring \( R_q = \mathbb{Z}_q / (x^n + 1) \). Then, the NTT transform of each coefficient of \( a(x) \) and its inverse NTT (INTT) with a primitive \( n\)-th root of

Fig. 1. Block diagram of a ring-LWE cryptosystem.
unity $\omega$ can be expressed as follows:

$$A_i = \text{NTT}_v^x(a(x)), = \sum_{j=0}^{n-1} a_j \omega^{jy} \text{mod } q$$  

(1)

$$a_i = \text{INTT}_v^x(A) = n \sum_{j=0}^{n-1} A_j \omega^{-jy} \text{mod } q$$  

(2)

The multiplication of two polynomials $a(x)$ and $b(x)$ is computed as (3), where $\circ$ denotes a point-wise multiplication, and two polynomials $a'(x)$ and $b'(x)$ are constructed as shown in (4).

$$c(x) = a(x) \cdot b(x) = \text{INTT}_v^y(\text{NTT}_v^x(a') \circ \text{NTT}_v^x(b'))$$  

(3)

$$a' = (a_0, \phi a_1, ..., \phi a_{n-1}), b' = (b_0, \phi b_1, ..., \phi b_{n-1})$$  

(4)

In [4], authors implemented an $n$-point NTT-core architecture using a systolic array to speed up the multiplication process compared with previous studies. However, this architecture is deployed using a single-path delay feedback (SDF), which offers a low throughput and efficiency. In addition, the SDF and multiple-path delay commutator (MDC) architectures deployed in recent researches [5, 6] are not sufficient to process the polynomial multiplication.

In this work, a multiple-path delay feedback (MDF) based NTT polynomial multiplier offering high efficiency with low latency is presented. By employing the NTT process of two polynomials $a(x)$ and $b(x)$ concurrently, the multiplication latency is reduced considerably. Additionally, the NTT core is designed using MDF architecture to achieve a better performance compared with previous polynomial multipliers.

**II. PROPOSED NTT POLYNOMIAL MULTIPLIER**

The block diagram of the proposed NTT polynomial multiplication is shown in Fig. 2. To conduct polynomial multiplication, the coefficients of input polynomials $a(x)$ and $b(x)$ are initially multiplied by a factor $\phi$, where $\phi \equiv \omega \text{mod } q$, to get the polynomials $a'(x)$ and $b'(x)$. These multiplication results are then transformed using NTT cores, followed by a point-wise multiplication and an INTT computation.

The polynomial multiplication result $c(x)$ is completely computed by multiplying the output of INTT with the factor $\phi^{-1}$. The entire operations of the multiplier are directed by a controller. To speed up the multiplication of the two polynomials $a(x)$ and $b(x)$, negative wrapped convolutions as well as NTT operations are controlled so that the multiplication latency is minimized. Specifically, the two NTT operations NTT($a'$) and NTT($b'$) are processed simultaneously, resulting in the reduction of one NTT operation compared with the architecture in [5]. The negative wrapped convolutions of the two input polynomials $a(x)$ and $b(x)$ are also conducted in parallel to decrease the processing time. Fig. 3 shows the timing diagram of the proposed NTT polynomial multiplier. As can be seen from Fig. 3, two computations $a \times \phi$ and $b \times \phi$ are scheduled to work simultaneously. Two NTT cores transforms NTT($a'$) and NTT($b'$) also work in parallel. Compared with the conventional architecture presented in [5], the proposed architecture offers a reduction in the processing time of one NTT computation and one negative wrapped convolution.

Fig. 4 presents the proposed NTT core using an 8-datapath MDF architecture to process the input data in parallel. Specifically, the coefficients of the input polynomial $a(x)$ are divided into 8 datapaths, which are processed by different processing elements (PE1s, PE2s). Each PE consists of modular reduction units (MR1,
MR2), constant multiplier, butterfly unit (BU), and one first-in-first-out (FIFO) register.

**III. RESULTS AND COMPARISONS**

The proposed NTT polynomial multiplier architecture is synthesized and implemented on a Xilinx Virtex-7 FPGA board. The simulation results and performance comparison are provided in Table 1. As can be seen from Table 1, the number of clock cycles necessary to complete a polynomial multiplication of the proposed architecture is only 226 clock cycles, which is about 6.25% and 8.69% of that of design 1 in [4] and the architecture in [5], respectively. In addition, the proposed architecture can operate at a higher clock frequency than others, resulting in a reduction in the multiplication latency. To compare the balance between area and latency of the proposed architecture with that of others, a parameter named area-latency product described in [4] is used. As presented in Table 1, the proposed polynomial multiplier architecture offers a much lower area-latency product compared with that of other architectures.

**IV. CONCLUSIONS**

An MDF-based NTT polynomial multiplier for ring-LWE cryptography is presented in this paper. By processing the NTT operation of the input polynomials concurrently, the multiplication process is speeded up. The implementation results prove that the proposed NTT polynomial multiplier achieves better performance than previous approaches. Therefore, it can be applied in ring-LWE and lattice-based cryptosystems to boost polynomial multiplication and improve encryption and decryption processes.

**ACKNOWLEDGMENTS**

This work was supported by the Basic Science Research Program (2019R1F1A1061926) through the NRF funded by the MSIT and by Inha University, Incheon, Korea.

**REFERENCES**


**Tuy Nguyen Tan** received the M.S. and Ph.D. degrees in Information and Communication Engineering from Inha University, South Korea, in 2016 and 2019, respectively. From 2009 to 2013, he had been a Telecommunication engineer at GTel Mobile JSC., Danang Branch, Vietnam. Since 2019, he has been a researcher at Faculty of Information Technology, Ton Duc Thang University, Vietnam. His research interest includes algorithm and VLSI architecture design for cryptosystems and error correction codes.

**Tram Thi Bao Nguyen** received the M.S. and Ph.D. degrees in Information and Communication Engineering from Inha University, South Korea, in 2016 and 2020, respectively. Since 2013, she has been with Danang University, Vietnam where she is a lecturer. Her research interests include VLSI architecture design for digital signal processing and communication systems.

**Hanho Lee** received the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Minnesota, Minneapolis, in 1996 and 2000, respectively. From 2000 to 2002, he was a Member of the Technical Staff with Lucent Technologies (Bell Labs Innovations), Allentown. From 2002 to 2004, he was an Assistant Professor with the Department of Electrical and Computer Engineering, University of Connecticut, USA. Since 2004, he has been with the Department of Information and Communication Engineering, Inha University, where he is currently a Professor. From 2010 to 2011, he was a Visiting Scholar with Bell Labs, Alcatel-Lucent, Murray Hill, NJ, USA. His research interests include algorithm and architecture design for cryptographic, forward error correction coding, and digital signal processing.