Systolic Acceleration of Polynomial Multiplication for KEM Saber and Binary Ring-LWE Post-Quantum Cryptography

Tianyou Bao*, Pengzhou He*, and Jiafeng Xie (Corresponding Author)
Department of Electrical and Computer Engineering, Villanova University, PA 19087; *: Equal contribution;
Email: {tbao,phe,jiafeng.xie}@villanova.edu
;

***Abstract*—Following the rapid progress in the post-quantum cryptography (PQC) field that many efforts have been gradually switched to the hardware implementation side, this paper presents a novel systolic accelerator for polynomial multiplication within two lattice-based PQC algorithms, key encapsulation mechanism (KEM) Saber and binary Ring-Learning-with-Errors (BRLWE)-based encryption scheme. Based on the observation that polynomial multiplication over ring is the key arithmetic operation for the two PQC schemes, we have proposed a novel systolic accelerator for the targeted polynomial multiplications (applicable to two PQC schemes). Mathematical formulation is given to illustrate the proposed algorithmic operation for both schemes. Then, the proposed systolic accelerator is presented. Finally, field-programmable gate array (FPGA) implementation results have been provided to confirm the efficiency of the proposed systolic accelerator under two schemes. The proposed accelerator is highly efficient, and the following work may focus on cryptoprocessor design and side-channel attacks.**

***Index Terms*—BRLWE-based scheme, KEM Saber, polynomial multiplication, PQC, systolic accelerator.**

I. Introduction

Post-quantum cryptography (PQC) has drawn significant attention from the research community recently as the existing public-key cryptosystems such as RSA (Rivest, Shamir, and Adleman) and elliptic curve cryptography (ECC) are proven to be vulnerable against quantum attacks [1], [2]. Overall, lattice-based cryptography, known for its strong security proof and relatively low implementation cost, has the solid potentiality to be deployed in emerging applications [2], [3].

Many of the lattice-based PQC algorithms are based on the Learning-with-Errors (LWE) problem or its variants [3]. Ring-LWE, a variant of LWE, is then proposed with reduced complexity [4]. After that, binary Ring-LWE (BRLWE, a variant of Ring-LWE) is proposed to build the ultra-lightweight PQC, BRLWE-based encryption scheme [5], as it uses binary errors to replace the regular Gaussian distributed ones. Meanwhile, Learning-with-Rounding (LWR) is also a variant of LWE [6], which has attracted a good number of works on the related PQC scheme [7], [8], including the National Institute of Science and Technology (NIST)'s 3rd round finalist, i.e., the key encapsulation mechanism (KEM) Saber [2].

On the other hand, not many hardware implementations have been released on the mentioned two PQC schemes, e.g., field-programmable gate array (FPGA) devices. As it is observed that polynomial multiplication over ring $\mathbb{Z}_q/(x^n+1)$ is the critical arithmetic operation for the mentioned two schemes (with different parameter settings) [5], [6], a uniform method can be developed to implement them efficiently.

Systolic structure has been deployed for many high-performance processor designing, due to its superior features such as high-throughput, high modularity, and regularity [8]. Systolic accelerators for PQC schemes, however, have not been reported yet. As the mentioned two PQC algorithms, KEM Saber and BRLWE-based encryption scheme, can be used in high-performance applications such as servers, the need to develop systolic accelerators for PQC is at an all-time high.

With this consideration, in this paper, we propose a novel systolic accelerator for polynomial multiplication of KEM Saber and BRLWE-based scheme (though with differences on the parameters). The major contributions of this work include:

- Delineating algorithmic derivations for the polynomial multiplication of targeted schemes for systolic processing.
- Presenting a novel systolic accelerator following optimized algorithm-to-architecture co-design strategies.
- Implementing the proposed accelerators on the FPGA platform (two schemes) and comparing them with the competing ones to demonstrate their efficiency.

Specifically, to the authors' best knowledge, this is the first systolic polynomial multiplication accelerator for KEM Saber and BRLWE-based PQC, which offers many unique features: (i) unified structure fits all security ranks; (ii) high-performance operation; and (iii) overall area-time efficiency.

The rest of this paper is organized as follows. Preliminaries are introduced in Section II. Algorithmic operation is formulated in Section III. The proposed systolic accelerator is provided in Section IV. Implementation and comparison are presented in Section V. Conclusions are given in Section VI.

II. Preliminaries

For simplicity of discussion, we just use the original notation to give the background information for two PQC schemes.

KEM Saber: Module LWR (MLWR)-based Encryption Scheme. As a variant of the LWE problem, LWR uses the errors that are produced from a rounding operation [6]. The samples are generated from $(\mathbf{a}, b = \lfloor \frac{p}{q}\langle \mathbf{a}, s\rangle \rceil_p) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$. The LWR with module matrices is called as MLWR.

KEM Saber is adaptive chosen ciphertext attack (IND-CCA2) secure, which is built on the hardness of the MLWR

problem [8]. Overall, KEM Saber has three phases: key generation, encapsulation, and decapsulation, see [6].

Module Ranks. KEM Saber has three module ranks named as LightSaber, Saber, and FireSaber, which correspond with NIST security levels 1, 3, and 5, respectively ($l = 2,\ 3,$ and 4). The sampled secrets are in the range of [-5,5] (LightSaber), [-4,4] (Saber), and [-3,3] (FireSaber), respectively [6]. Besides that, polynomial degree is $n = 256$ with $q = 2^{13}$ and $p = 2^{10}$.

BRLWE-based Encryption Scheme. BRLWE-based PQC involves three phases: key generation, encryption, and decryption. As shown in [5], one can conclude that the major arithmetic operation of the BRLWE-based PQC is the polynomial multiplication (followed by a polynomial addition), where one polynomial consists of integer coefficients and another polynomial has merely binary coefficients (see details in [5]).

Inverted Range Representation. [10] has proposed using the inverted range representation $(-\lfloor q/2 \rfloor, \lfloor q/2 \rfloor - 1)$ for the BRLWE-based scheme, to facilitate using two's complement format. We also follow this strategy here.

Security of the BRLWE-based Scheme. BRLWE-based scheme is based on the average-case hardness of the BRLWE problem [5]. The BRLWE-based PQC achieves 73-bits & 140-bits quantum security for (n, q)=(256,256) and (n, q)=(512,256), respectively, which fits well lightweight applications [11].

Prior Works for KEM Saber. There exist two types of hardware implementations for KEM Saber, namely the hardware-software co-design and full hardware design. The former type of designs can be seen at [12] and [13], respectively. While for the latter type, the first hardware implementation was presented in [14]. A Karatsuba based hardware Saber was reported in [15]. Efficient polynomial multiplication structures for KEM Saber were then presented in [16]. A compact KEM Saber processor was proposed [17]. High-performance KEM Saber polynomial multiplications were given in [18].

Prior Works for BRLWE-based Scheme. The first software implementation of the BRLWE-based scheme was reported in [5]. The first hardware BRLWE-based PQC was presented in [19]. This work was then followed by a pair of high-speed and ultra-lightweight structures in [10]. An efficient high-speed BRLWE-based PQC architecture was proposed in [20] recently. A new compact design was reported in [21].

Polynomial Multiplication for KEM Saber. Polynomial multiplication of KEM Saber has the setup as: one polynomial has small-size coefficients (e.g., [-4,4] for Saber) and the other one has coefficients of 10-/13-bit (we use 13-bit here) [6].

Polynomial Multiplication for BRLWE-based Scheme. One polynomial has only binary values while another one has coefficients of 8-bit ($\log_2 q = \log_2 256 = 8$) [5].

III. Algorithmic Derivation

General Notation Definition. Without loss of generality, we can define a general polynomial multiplication for KEM Saber and BRLWE-based scheme as $(f(x) = x^n + 1)$

$$W = BD \bmod f(x), \tag{1}$$

where $W = \sum_{i=0}^{n-1} w_i x^i$, $B = \sum_{i=0}^{n-1} b_i x^i$, and $D = \sum_{i=0}^{n-1} d_i x^i$, where b_i is the small-size coefficient (e.g., [-4,4] for Saber) and d_i and w_i are larger-size coefficients over ring (e.g., 8-bit for BRLWE-based scheme). Note that an additional integer polynomial $G = \sum_{i=0}^{n-1} g_i x^i$ is needed for the operations within BRLWE-based PQC, following [10].

We then have $W = \sum_{i=0}^{n-1} d_i (Bx^i \bmod f(x))$, which is

$$\begin{aligned} W = &\quad w_0 \quad + \quad w_1 x \quad + \cdots + \ w_{n-1} x^{n-1} \\ =&(\quad b_0 d_0 \quad + \quad b_1 d_0 x \quad + \cdots + b_{n-1} d_0 x^{n-1}) \\ &+ \cdots \ \cdots \ \cdots \\ &+ (-b_1 d_{n-1} - b_2 d_{n-1} x - \cdots + b_0 d_{n-1} x^{n-1}), \end{aligned} \tag{2}$$

where $x^n \equiv -1$ is substituted ($x^n + 1 \equiv 0$). We then have

$$\begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{n-1} \end{bmatrix} = \begin{bmatrix} b_0 & -b_{n-1} & \cdots & -b_1 \\ b_1 & b_0 & \cdots & -b_2 \\ \vdots & \vdots & \ddots & \vdots \\ b_{n-1} & b_{n-2} & \cdots & b_0 \end{bmatrix} \times \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{n-1} \end{bmatrix}, \tag{3}$$

which can be $[W] = [B] \times [D]$. Based on (3), we can define each element within matrix $[B]$ as $[B]_{i,j}$ $(1 \le i, j \le n)$, e.g., $[B]_{1,1} = b_0$ and $[B]_{1,n-1} = -b_1$ (similarly, $[W]_{j,1}$ and $[D]_{j,1}$ refer to an element in the vector). For $n = uv$ (u and v are integers), we can have the proposed algorithmic operation as:

Algorithm 1: Proposed algorithmic operation for polynomial multiplication (KEM Saber and BRLWE-based PQC scheme)

Input : B, D, and G (G is only for BRLWE-based scheme; B has small-size coefficients; D (or G) has larger-size coefficients;

Output: $W = BD \bmod f(x)$ $(f(x) = x^n + 1)$; // G is also added for BRLWE-based PQC

Initialization step

Make ready B and D (or G);

$Z = 0$;

Main step

$Z = [G]$; // this operation only applies to BRLWE-based PQC

for $j = 1$ *to* v **do**

  for $i = 1$ *to* n **do**

    $Z = Z + \sum_{k=1}^{u} [B]_{i,ju+k} [D]_{iu+k,1}$;

  end

end

$W = Z$;

Final step

Deliver the output W;

Note that during actual implementation, we assume the elements within $[B]$ are l_1-bit (including sign) and the elements of $[D]$ and $[W]$ (even $[G]$) are l_2-bit (with sign involved), e.g., $l_1 = 2$ when BRLWE-based encryption scheme is applied.

IV. Proposed Systolic Accelerator

Following Algorithm 1 of Section II, we can have the proposed hardware accelerator for the major arithmetic operation of KEM Saber and BRLWE-based PQC as shown in Fig. 1 (general form). The proposed accelerator contains u number of processing elements (PEs), one accumulation cell (AC), and

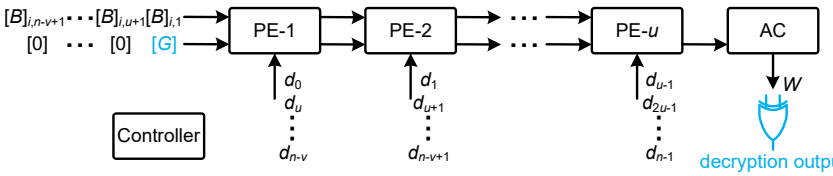

Fig. 1. Proposed systolic accelerator for polynomial multiplication (KEM Saber and BRLWE-based scheme), where the blue highlighted components are only applicable to BRLWE-based PQC. PE: processing element. AC: accumulation cell. Note that all the $[B]$ inputs are represented in the sign magnitude format.

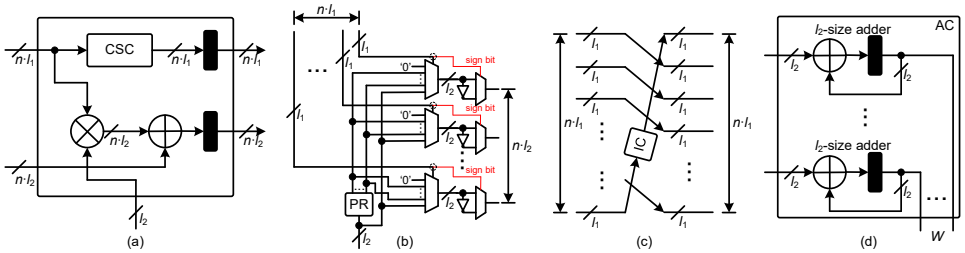

Fig. 2. Internal structure of (a) PE (black box is the register); (b) multiplier; (c) circular-shifting cell (CSC); IC (inverter cell); (d) AC.

one controller. Note that all the $[B]$ inputs are represented in the sign magnitude format (other inputs/outputs use the two's complement). These components are described as follows.

PE. The internal structure of the PE is shown in Fig. 2(a), where it has one circular-shifting cell (CSC), one multiplier cell, one adder cell, and two register cells. In total, $n \cdot l_1$ bits from B are fed to the PE (attached to the multiplier cell and CSC). The multiplier cell has n parallel MUXes, where each MUX is fed with all the pre-computed point-wise values. Benefited from the sign magnitude representation format, only positive results and '0' are included here through the pre-computation (PR) cell. For instance, for BRLWE-based scheme, only '0' and 'X' values are needed for each MUX (assume 'X' is the value of the l_2-bit input from the bottom). Similar strategy applies to the case of KEM Saber, e.g., '2X', '3X', and '4X' will be calculated from PR cell (when the l_1-bit input lies in [-4,4]). The adder cell has n number of l_2-bit full adders working in parallel to produce $n \cdot l_2$ bits (adding another input of the PE). The detail of the CSC is shown in Fig. 2(c), where it circularly shifts the input by one position with one value being inverted through the inverter cell (IC). The IC contains 1 NOT-gate (based on sign magnitude format). The output of the CSC and the adder cell are then fed to the following $n \cdot l_1$- and $n \cdot l_2$-size registers, respectively. Note that PE-u does not have CSC since it connects with AC directly.

AC. The AC contains n number of l_2-size adders followed by n number of l_2-size registers (Fig. 2(d)). The AC functions to accumulate the sequentially computed results from these PEs to produce the final output according to Algorithm 1. Note that for the BRLWE-based scheme, the output of AC ($n \cdot l_2$-size) is also attached to n XOR gates to generate the decryption output (each XOR is connected with two most significant bits (MSBs) of a l_2-bit output coefficient).

Controller. A controller is needed for the systolic accelerator to execute the algorithmic operation in Algorithm 1. The controller is based on a finite state machine (FSM) to produce all necessary signals for the operations of the accelerator.

Overall operation. The two inputs B and G (G is only for the BRLWE-based scheme) are fed to the accelerator according to the format shown in Fig. 1. While the coefficients

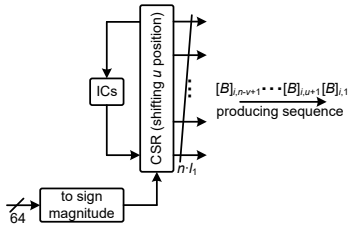

Fig. 3. Proposed CSR for producing all $[B]$ inputs.

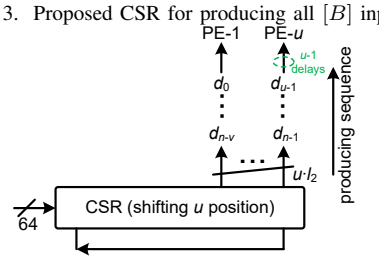

Fig. 4. Proposed CSR for producing all $[D]$ inputs.

of D are accordingly divided into several groups to be fed to the respective PEs (matching corresponding $[B]_{i,ju+k}$). PE-u delivers its first output to AC after u cycles (after $[B]_{i,1}$ or $[G]$ are fed to PE-1) and will keep delivering results to AC in the following cycles. The accumulation takes v cycles and then all the output coefficients are available in parallel (decryption output is also available for the BRLWE-based PQC).

V. Implementation and Comparison

For a detailed complexity analysis of the proposed systolic accelerator (applicable to both KEM Saber and BRLWE-based scheme), we have implemented the corresponding polynomial multiplication accelerators for both PQC schemes on the FPGA platform. The experimental setup is as follows:

(i) With respect to the practical application environment, we have set that the input/output data are read/write from/to a 64-bit RAM. Besides that, we have used circular-shift registers (CSRs) to produce related $[B]$ and $[D]$ inputs (see Fig. 3 and Fig. 4). Meanwhile, an output buffer is also attached to the AC of Fig. 1 such that the output is delivered back to the 64-bit RAM serially and the delivery time is matched with input loading time (for the sake of systolic processing).

(ii) The proposed systolic accelerators for KEM Saber ($n = 256$, $l_1 = 4$ as [-4,4], $l_2 = 13$) and BRLWE-based PQC of $(n, q) = (256, 256)$ and $(n, q) = (512, 256)$ ($l_1 = 2, l_2 = 8$)

TABLE I
COMPARISON OF FPGA IMPLEMENTATION RESULTS FOR DIFFERENT BRLWE-BASED PQC ACCELERATORS/STRUCTURES

design	n	phase	device	u	ALMs	Fmax(MHz)	latency[1]	delay	ADP[2]	ADP reduction
[20]	256	Dec.	Stratix-V	-	4,495	321.03	258	0.804	3,612	-
Prop. Work	256	Dec.	Stratix-V	4	13,211	277.32	68	0.245	3,239	10.32%
				8	25,212	277.32	40	0.131	3,298	8.70%
[20]	512	Dec.	Stratix-V	-	9,038	317.06	514	1.621	14,651	-
Prop. Work	512	Dec.	Stratix-V	4	26,368	253.81	132	0.520	13,713	6.40%
				8	50,285	249.19	72	0.289	14,529	0.83%

The recent high-speed design of [20] is listed here for comparison ([21] is a compact design with low-speed). Unit for delay (critical-path×latency): μs. Dec.: decryption phase. [1]: Latency cycles (decryption phase), where CSRs' loading and final output delivery time are not included. [2]: ADP=#ALM×delay.

TABLE II
COMPARISON OF FPGA IMPLEMENTATION RESULTS FOR DIFFERENT KEM SABER-BASED POLYNOMIAL MULTIPLICATION STRUCTURES

design	n	device	u	ALMs	LUT	FF	Slice	Fmax (MHz)	latency[1]	delay	ADP[2]	ADP reduction[3]
[16] (HS-I)	256	Ultrascale+	-	-	10,844	5,150	-	250.00	256	1.02	11.10	1.9%
[16] (HS-II)	256	Ultrascale+	-	-	22,118	4,920	-	250.00	128	0.51	11.32	-
[18] (Fig.7)	256	Stratix-V	-	14,341	-	-	-	204.54	128	0.63	8.97	-
Prop. Work	256	Stratix-V	4	19,782	-	-	-	287.6	68	0.24	4.68	47.9%
			8	32,989	-	-	-	278.4	40	0.14	4.74	47.2%
Prop. Work	256	Ultrascale+	4	-	29,450	24,603	4,821	250.00	68	0.27	8.01	29.2%
			8	-	49,640	38,210	8,659	250.00	40	0.16	7.94	29.8%

Unit for delay (critical-path×latency): μs. [1]: Latency cycles, not including CSRs' loading and final output delivery time.
[2]: For Stratix-V: ADP=#ALM×delay; For Ultrascale+: ADP=#LUT×delay. [3]: Calculation is based on the same FPGA device.

are coded with VHDL and implemented on different FPGA devices for $u = 4$ and $u = 8$. The obtained results are shown in Table I and II, respectively, along with the existing designs.

The proposed accelerators have significantly better performance than the existing designs, even though the proposed accelerator has included complete input/output processing components (which is missing in the existing designs). The proposed accelerator (KEM Saber) achieves at least 29.2% less area-delay product (ADP) than the state-of-the-art designs [16], [18]. Similarly, the proposed accelerator (BRLWE-based PQC) also has better ADP than [20]. Besides, as the proposed accelerator has matched input & output processing time, the actual performance (processing successively) is much better than the existing designs due to systolic operation.

Overall, the proposed accelerators have features of: (i) applicable to two PQC schemes and different security levels; (ii) high-performance operation; (iii) overall area-time efficiency. Due to the adding of practical input/output processing components, the proposed accelerators may have slightly slower frequency than the existing ones (e.g., Table I). Nevertheless, our reported data are based on practical setup and can be considered as actual results under high-performance operations.

This is the first report on the systolic polynomial multiplication accelerator for KEM Saber and BRLWE-based PQC. The following work may focus on extending the accelerator to actual cryptoprocessor design and related side-channel attacks.

VI. CONCLUSION

This paper, for the first time, has proposed a novel systolic accelerator for polynomial multiplication within KEM Saber and BRLWE-based PQC. The related algorithmic operation is proposed and then mapped into a novel systolic accelerator. Following implementation and comparison have demonstrated the superior performance of the proposed accelerator.

VII. ACKNOWLEDGEMENT

J. Xie is supported by NSF SaTC-2020625 and NIST-60NANB20D203.

REFERENCES

[1] D. Bernstein. Introduction to post-quantum cryptography. *PQC*, 2009.
[2] Post-quantum cryptography round 3 submissions. https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions
[3] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM*, vol. 56, no. 6, 34, 2009.
[4] V. Lyubashevsky et al., "On ideal lattices and learning with errors over rings," *Eurocrypt*, 2010.
[5] J. Buchmann et al., "High-performance and lightweight lattice-based public-key encryption," *ACM Int. IoT Privacy, Trust, and Security*, 2016.
[6] J.-P. D'Anvers et al., "Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM," *AFRICACRYPT*, 2018.
[7] A. Karmakar et al., "Saber on ARM CCA-secure module lattice-based key encapsulation on ARM," *IACR Cryptology ePrint*, 2018:682, 2018.
[8] J. Mera et al., "Time-memory trade-off in Toom-Cook multiplication: an Application to Module-lattice based cryptography," *IACR TCHES*, 2020.
[9] H.-T. Kung. Why systolic architectures? *IEEE computer*, vol. 15, no. 1, pp. 37–46, 1982.
[10] S. Ebrahimi et al., "Post-quantum cryptoprocessors optimized for edge and resource-constrained devices in IoT," *IEEE IoT-J*, 2019.
[11] F. Göpfert et al., "A hybrid lattice basis reduction and quantum search attack on LWE," *PQCrypto*, 2017.
[12] J. Mera et al., "Compact domain-specific co-processor for accelerating module lattice-based KEM," *DAC*, pp. 1-6, 2020.
[13] V. Dang et al., "Implementing and benchmarking three lattice-based post-quantum cryptography algorithms using software/hardware codesign," *FPT*, 2019.
[14] S. Roy et al., "High-speed instruction-set coprocessor for lattice-based key encapsulation mechanism: Saber in hardware," *IACR TCHES*, 2020.
[15] Y. Zhu et al., "LWRpro: An energy-efficient configurable crypto-processor for Module-LWR," *IEEE TCAS-I*, 2021.
[16] A. Basso and S. Sinha Roy, "Optimized polynomial multiplier architectures for post-quantum KEM Saber," *DAC'21*, pp.1-6, 2021.
[17] P. He et al., "Compact coprocessor for KEM Saber: novel scalable matrix originated processing," *NIST Third PQC Stand. Conf.*, pp. 1-16, 2021.
[18] J. Xie et al., "CROP: FPGA implementation of high-performance polynomial multiplication in Saber KEM based on novel cyclic-row oriented processing strategy," *ICCD*, pp. 1-8, 2021.
[19] A. Aysu et al., "Binary Ring-LWE hardware with power side-channel countermeasures. *DATE*, pp. 1253-1258, 2018.
[20] J. Xie et al., "Efficient Implementation of finite field arithmetic for Binary Ring-LWE post-quantum cryptography through a novel Lookup-table-like method. *DAC*, 2021.
[21] P. He et al., "Novel low-complexity polynomial multiplication over hybrid fields for efficient implementation of Binary Ring-LWE post-quantum cryptography. *IEEE JETCAS*, 2021.