Work-in-Progress: High-Performance Systolic Hardware Accelerator for RBLWE-based Post-Quantum Cryptography

Tianyou Bao[1], José L. Imaña[2], Pengzhou He[1], and Jiafeng Xie[1]
[1]: Dept. Electrical & Computer Engineering, Villanova University, USA. Email: {tbao,phe,jiafeng.xie}@villanova.edu.
[2]: Dept. Computer Architecture & Automation, Complutense University, Madrid, Spain. Email: jluimana@ucm.es.

***Abstract*—Ring-Binary-Learning-with-Errors (RBLWE)-based post-quantum cryptography (PQC) is a promising scheme suitable for lightweight applications. This paper presents an efficient hardware systolic accelerator for RBLWE-based PQC, targeting high-performance applications. We have briefly given the algorithmic background for the proposed design. Then, we have transferred the proposed algorithmic operation into a new systolic accelerator. Lastly, field-programmable gate array (FPGA) implementation results have confirmed the efficiency of the proposed accelerator.**

***Index Terms*—Polynomial multiplication, PQC, RBLWE, systolic hardware accelerator**

I. Introduction

Post-quantum cryptography (PQC) has drawn significant attention from the research community recently [1-2]. While the ongoing National Institute of Science and Technology (NIST) PQC standardization process targets general-purpose PQC algorithms, there is also a need of developing lightweight PQC for related applications such as Internet-of-Things (IoT) devices/servers. Ring-Binary-Learning-with-Errors (RBLWE, a variant of Ring-LWE)-based scheme is a promising PQC to serve such a role as it uses binary errors to obtain small computational complexity. Several related works have recently been carried out on this PQC scheme [3-8].

Though a few designs for RBLWE-based PQC have been released, high-performance hardware acceleration (e.g., for IoT servers) for the RBLWE-based scheme has not been well covered. Meanwhile, the recent research on the PQC field has gradually switched to the hardware implementation side [8]. In particular, it is noticed that the recent high-speed structures for RBLWE-based scheme are mostly based on the processing setups of serial-in serial-out or parallel-in serial-out [8]. This type of input/output processing, however, may not be ideal for high-performance applications as the targeted environments like IoT servers actually have abundant resources for deploying higher-speed PQC accelerators. With these considerations, in this paper, we propose to design a novel hardware systolic accelerator for RBLWE-based PQC under such applications.

II. Preliminaries

Notation 1. Define n is the security level of the RBLWE-based PQC and the size of the polynomial (over ring $\mathbb{Z}_q/(x^n+1)$) and $\log_2 q$ as the bit-width for the coefficients in the integer polynomial [3].

Overview. RBLWE-based PQC scheme involves three phases: key generation, encryption and decryption [3]. As shown in Fig. 1, one can conclude that the major arithmetic operation of the RBLWE-based PQC is the polynomial multiplication

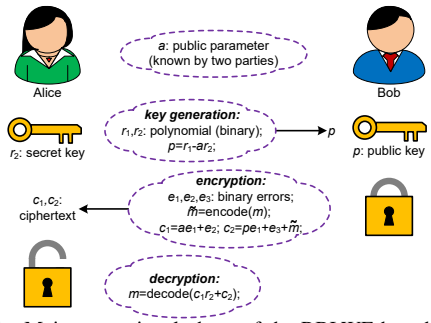

Fig. 1. Major operational phase of the RBLWE-based PQC.

(followed by a polynomial addition), where one polynomial consists of integer coefficients and another polynomial involves merely binary coefficients.

Inverted range representation. A recent report [5] has used the inverted range representation $(-\lfloor q/2 \rfloor, \lfloor q/2 \rfloor - 1)$ for the integer coefficients of the polynomial such that all the involved modular addition/subtraction can be performed without any reduction. We also follow this strategy for the proposed design.

Quantum security. The RBLWE-based scheme is based on the average-case hardness of the RBLWE problem [3], which achieves 73-bits/140-bits quantum security for (n, q)=(256,256) and (n, q)=(512, 256), respectively (fits well lightweight applications [6]).

III. Algorithmic Operation

Following Section 2, one can use the operation in the decryption phase as the typical algorithmic operation for RBLWE-based PQC.

Notation 2. Define the major algorithmic operation as

$$W = DB + G \bmod f(x) = DB + G \bmod (x^n + 1), \quad (1)$$

where $W = \sum_{i=0}^{n-1} w_i x^i$, $D = \sum_{i=0}^{n-1} d_i x^i$, $G = \sum_{i=0}^{n-1} g_i x^i$, and $B = \sum_{i=0}^{n-1} b_i x^i$; b_i denotes the binary coefficient and d_i, g_i and w_i are $\log_2 q$-bit coefficient over $\mathbb{Z}_q$. Generally, (1) can be transferred into

$$\begin{bmatrix} w_0 \\ \vdots \\ w_{n-1} \end{bmatrix} = \begin{bmatrix} d_0 & \cdots & -d_1 \\ \vdots & \ddots & \vdots \\ d_{n-1} & \cdots & d_0 \end{bmatrix} \times \begin{bmatrix} b_0 \\ \vdots \\ b_{n-1} \end{bmatrix} + \begin{bmatrix} g_0 \\ \vdots \\ g_{n-1} \end{bmatrix}, \quad (2)$$

where we define each element within matrix $[D]$ as $[D]_{i,j}$, e.g., $[D]_{1,1} = d_0$ and $[D]_{1,n-1} = -d_1$. Similarly, we have $[W]_{0,1} = w_0$ (also $[B]$ and $[G]$). We can have the proposed algorithm in Fig. 2.

Algorithm 1: Proposed algorithmic operation for RBLWE-based PQC

Input: B, D, and G (B is a binary polynomial, D and G are $\log_2 q$-bit integer polynomials);
Output: $W=BD+G \bmod f(x)$ ($f(x)=x^n+1$);
Initialization step
1 Make ready B, D, and G;
2 Z=0;
Main step
3 Z=[G];
4 **for** j=1 to n do
5 | **for** i=1 to n do // parallel execution
6 | | $Z=Z+[B]_{i,j}[D]_{j,1}$;
7 | **end**
8 **end**
9 W=Z;
Final step
10 Deliver the output W;

Fig. 2. Major operational phase of the RBLWE-based PQC.

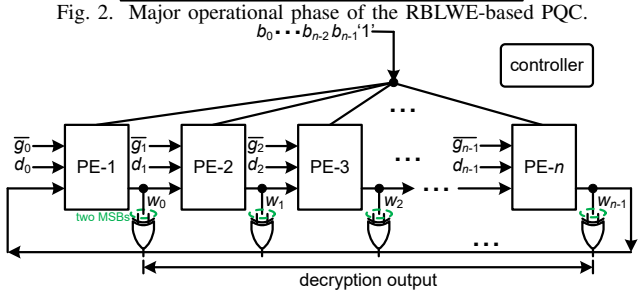

Fig. 3. Proposed systolic accelerator, where $\overline{g_i}$ denotes the inverted g_i (two's complement form). PE: processing element.

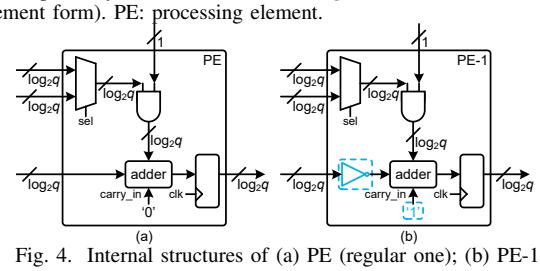

Fig. 4. Internal structures of (a) PE (regular one); (b) PE-1.

IV. Proposed Systolic Accelerator

Following Algorithm 1, we can have the proposed hardware accelerator of Fig. 3. The components are described below.

PE. The internal structure of the PE is shown in Fig. 4(a). The signals of $\overline{g_i}$ and d_i are fed to the MUX first and then connected to the following AND. The AND cell has n parallel AND cells (each has $\log_2 q$ AND gates), where one inputs of these AND gates are connected with the 1-bit input from top and another input of these AND gates are connected with these $\log_2 q$-bit (from MUX), respectively. Note that PE-1 needs an extra inverter of $\log_2 q$-bit (the carry_in of the adder is also set as '1') to meet the requirement of two's complement [8], as highlighted in Fig. 4(b). Besides that, all the coefficients of G are needed to be inverted (see Fig. 3) [8].

Controller. A controller is needed for the operation of the accelerator. The controller is based on a finite state machine, which involves three stages, namely "loading", "computation", and "done".

Overall operation. The two inputs D and G are fed to the accelerator according to Line 1 of Algorithm 1. While the coefficients of G (inverted form) are loaded into the respective PEs when the first input on top is '1'. Then, all coefficients of D are multiplied with matched coefficients of B in a serial format, and results are accumulated through the paired adder and register in the PEs. The overall loading and computation take $(n+1)$ cycles, and then the output becomes available in parallel (n parallel XORs are connected with the output's two most significant bits (MSB) [4]).

TABLE I
Comparison of the Implementation Results

design	n	ALMs	Fmax	latency	delay[1]	ADP*
[5]	256	5,734	369.14	257	0.696	3,991
[7]	256	4,495	321.03	258	0.804	3,614
Prop.	256	5,271	414.25	257	0.62	3,268
[5]	512	11,470	336.36	513	1.525	17,492
[7]	512	9,038	317.06	514	1.621	14,652
Prop.	512	10,525	368.6	513	1.391	14,648

Unit for delay: ns. *: ADP=#LUT×delay ($\times 10^3$). [1]: delay is calculated as latency × (1/Fmax), where the latency refers to the computation time (Decryption phase).
For a fair comparison, we have only compared with the existing designs of [5], [7] with similar input and output setup, i.e., parallel-in and/or parallel-out.

V. Complexity and Comparison

We have implemented the design of Fig. 3 on the field-programmable gate array (FPGA) platform. The proposed accelerator is coded with VHDL and verified by ModelSim, and we have selected Intel Stratix-V 5SGXMABN1F45C2 device (follow [7]) to obtain implementation results through Intel Quartus Prime 17.0. We have listed the adaptive logic modules (ALMs), maximum frequency (Fmax), latency cycles, delay (critical-path × latency, where critical-path=1/Fmax), and area-delay product (ADP= #ALM×delay) in Table 1. One can see that the proposed design has smaller area-time complexities than the existing ones [5,7], e.g., for n=256, the proposed design has 9.6% less ADP than [7] (similar to $n = 512$).

Discussion. The proposed accelerator still needs development: (i) novel algorithmic/ architectural innovations are needed to design a more efficient accelerator; (ii) related side-channel attacks and countermeasures are also required to be developed.

VI. Conclusion

An efficient hardware systolic accelerator for RBLWE-based PQC is proposed. We have proposed three layers' efforts. The following work may focus on algorithmic/architectural innovations and side-channel attacks and countermeasures.

Acknowledgment

J. Xie was supported by NSF SaTC-2020625 and in part by NIST-60NANB20D203. J.L. Imaña was supported by PID2021-123041OB-I00 funded by MCIN/AEI/ 10.13039/501100011033 and by "ERDF A way of making Europe", and by the CM under grant S2018/TCS-4423.

References

[1] W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. *Symp. Founda. of Computer Science*, pp. 124-134, 1994.
[2] Post-quantum cryptography round 3 submissions. https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions
[3] J. Buchmann et al., "High-performance and lightweight lattice-based public-key encryption," *ACM IoT Priv., Trust, & Security*, 2016.
[4] A. Aysu et al., "Binary Ring-LWE hardware with power side-channel countermeasures," *DATE*, pp. 1253-1258, 2018
[5] S. Ebrahimi et al., "Post-quantum cryptoprocessors optimized for edge and resource-constrained devices in IoT," *IEEE J-IoT*, 2019.
[6] F. Göpfert et al., "A hybrid lattice basis reduction and quantum search attack on LWE," *PQCrypto*, pp. 184-202, 2017.
[7] 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, pp. 1279-1284.
[8] J. Imaña et al., "Efficient Hardware Arithmetic for Inverted Binary Ring-LWE Based Post-Quantum Cryptography," *IEEE TCAS-I*, 2022.