Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.081651

ARTICLE

Enhancing Efficiency in Lattice-Based Post-Quantum Cryptography with Systolic Array Polynomial Multiplication

Atef Ibrahim[1,*] **and Fayez Gebali**[2]

[1]Computer Engineering Department, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia

[2]Electrical and Computer Engineering Department, University of Victoria, Victoria, BC, Canada

*Corresponding Author: Atef Ibrahim. Email: aa.mohamed@psau.edu.sa

Received: 10 March 2026; Accepted: 20 May 2026; Published: 23 July 2026

ABSTRACT: The ongoing expansion of the Internet of Things (IoT) fundamentally alters industrial and economic paradigms by integrating intelligent nodes throughout operational frameworks. Nonetheless, vulnerabilities surrounding system integrity and data confidentiality present major bottlenecks to widespread adoption, a dilemma severely intensified by impending quantum computing capabilities. Defending these networks demands the integration of post-quantum cryptographic primitives; yet, the severe hardware constraints characterizing peripheral IoT components complicate practical deployment. Quantum-resistant lattice cryptography offers a highly promising pathway to overcome these limitations, largely because the foundational security and throughput of these protocols hinge on polynomial multiplication performance. Consequently, optimizing the computational speed and architectural efficiency of this specific algebraic operation drastically enhances the viability of lattice-reliant defense mechanisms. To address this need, this study develops a specialized systolic array architecture engineered explicitly as an underlying arithmetic engine for polynomial multiplication within the Binary Ring Learning With Errors (BRLWE) protocol. Tailored for low-power hardware security modules (HSMs) situated at the network edge, the proposed circuit achieves rapid modular multiplication while ensuring a highly compact silicon footprint. By aligning the hardware layout with the precise algebraic properties of the BRLWE variation, this approach delivers a scalable, optimized framework for constructing secure IoT networks capable of resisting quantum adversaries, thereby acting as a pivotal building block for resilient industrial edge protection. Additionally, this study aligns with UN Sustainable Development Goals 8 and 9 by fostering digital trust in emerging technological systems and supporting the safe, adaptive growth of modern electronic economies.

KEYWORDS: Safeguarding IoT networks; post-quantum cryptography; IoT edge computing; systolic computation structures; lattice-based security techniques; multiplication of polynomials; sustainable development goals; future economies

1 Introduction and Related Work

The pervasive deployment of IoT systems is fundamentally transforming daily environments and industrial operations through interconnected networks of sensors and distributed computational platforms. These systems support continuous, real-time data acquisition and local processing, enabling intelligent and responsive service delivery. For example, in smart healthcare, wearable IoT devices monitor patient vitals and enable remote diagnostics, allowing for timely medical intervention and reducing hospital visits. Within industrial automation, embedded sensors track equipment performance and environmental conditions,

supporting predictive maintenance and optimizing production workflows. By enhancing operational efficiency and enabling innovative service models, IoT ecosystems serve as significant enablers of modern economic advancement. This aligns with Sustainable Development Goal (SDG) 9, which emphasizes resilient infrastructure and inclusive innovation, while also advancing SDG 8 through technology-driven, sustainable economic growth.

Nevertheless, expanding IoT adoption introduces substantial security and privacy vulnerabilities that can erode user trust and impede wider implementation. As critical functions increasingly shift to distributed edge nodes, these devices become attractive targets for sophisticated cyber threats, a risk further heightened by the emerging potential of quantum computing. Successful attacks could result in significant operational disruptions, financial harm, and delays in the transition toward secure digital economies [1].

Addressing these vulnerabilities requires cryptographic solutions that are both efficient enough for resource-constrained hardware and resilient against future quantum attacks. Developing secure, high-performance systems within these constraints is essential for establishing a trustworthy and sustainable digital infrastructure.

Standard IoT frameworks generally allocate processing tasks across a hierarchy of centralized cloud servers, intermediate fog or edge systems, and diverse client-end devices, where each tier operates under distinct processing, power, and dimensional constraints [2]. Centralized clouds and advanced edge stations often leverage high-performance processors or specialized hardware engines; conversely, peripheral end nodes are bound by rigid constraints regarding storage capacity, processing bandwidth, power availability, and architectural volume. To provide robust protection within these strict thresholds, custom hardware designs for asymmetric cryptography can be engineered, allowing specialized circuits to be finely tuned for specific performance indexes, including execution delay, operational throughput, power consumption, and chip area [3].

Despite the continued dominance of widespread public-key frameworks like ECC and RSA, their foundational cryptographic assumptions are severely threatened by quantum-computational algorithms, specifically Shor's algorithm. Furthermore, even within conventional computing paradigms, deploying ECC or RSA on edge nodes frequently introduces substantial operational overhead because of extensive key lengths and intricate mathematical computations. Relying on these traditional primitives for durable IoT infrastructures consequently introduces future-proofing security vulnerabilities alongside immediate deployment difficulties in the post-quantum epoch [1].

Quantum-resistant cryptography explores alternative paradigm structures that rely on underlying mathematical complexities thought to be unbreakable by both classical and quantum-scale attackers. The primary paradigms encompass code-based, lattice-based, multivariate quadratic, and isogeny-driven frameworks [4,5]. Out of these methodologies, lattice-reliant mechanisms have emerged as leading candidates due to their robust theoretical proofs, compact parameter footprints, and processing speed—attributes that align exceptionally well with low-resource IoT infrastructures. More structured variations, such as Ring-Learning with Errors (Ring-LWE), utilize algebraic operations over polynomial rings to scale down memory usage and computational burdens, thereby enhancing their feasibility for edge-level implementation [2,6].

The significance of lattice-based security has been heavily reinforced by recent global standardization initiatives spearheaded by NIST. Algorithms like CRYSTALS-Kyber for key encapsulation mechanisms and CRYSTALS-Dilithium for cryptographic signatures have been designated as foundational standards [7–9]. Concurrently, alternative lattice-driven designs such as NTRU and SABER demonstrate excellent performance profiles, driving their integration into IoT-centric security architectures. For energy-restricted,

battery-operated edge components, these breakthroughs encourage the creation of customized hardware co-processors engineered to exploit the distinct algebraic properties of lattice equations to streamline modular and polynomial arithmetic operations [1].

In the broader LWE landscape, the foundational version provides mathematically proven security reductions tied directly to worst-case lattice configurations. Modifications like Ring-LWE utilize specific polynomial rings to substantially minimize parameter dimensions without compromising security guarantees, which paves the way for resource-efficient hardware integration [4]. Additional structural modifications, such as utilizing ternary or binary noise distribution models, serve to optimize the sampling mechanics and lower storage footprints, producing lightweight variations that are highly compatible with ultra-low-power IoT hardware [10].

Binary Ring-LWE (BRLWE) employs binary error terms, simplifying sampling and lowering computational overhead, which makes it especially attractive for energy-constrained devices [11]. Concurrently, the deterministic Learning with Rounding (LWR) problem eliminates explicit error sampling and underpins efficient designs like Saber, a key encapsulation mechanism that has progressed through NIST's standardization process and is optimized for hardware efficiency [12].

A vast body of academic work has investigated the hardware realization of Ring-LWE and Saber using FPGA and ASIC technologies, examining concurrent structures for executing polynomial multiplication and associated tasks [13–17]. Evidence suggests that meticulously engineered processing paths and memory frameworks can drastically boost data processing rates and power economy relative to software-based approaches, rendering post-quantum cryptographic methods feasible for real-world edge applications. Because of their consistent data movement and extensive pipelining, systolic arrays provide specific benefits for high-speed polynomial calculations. Despite these strengths, specialized systolic-based engines for protocols such as Saber and BRLWE have received less attention than traditional parallel architectures [13,15].

Within the domain of BRLWE-reliant IoT deployment, the execution of polynomial multiplication acts as the principal processing bottleneck, highlighting the critical demand for dedicated, highly efficient computational architectures. This study centers on engineering a refined hardware-level multiplier intended to operate as the core arithmetic engine for Ring-LWE alongside its binary variants. Utilizing a unified algorithm-to-architecture co-optimization methodology, the internal multiplication routine is rearranged to yield uniform, localized dataflow paths that map perfectly onto a systolic array topology. The resulting FPGA-based implementation functions as a specialized cryptographic coprocessor that enhances both processing throughput and silicon real estate utilization, providing an essential building block for quantum-resistant IoT frameworks.

Prior literature focusing on silicon-level multipliers for lattice-reliant cryptography provides a vital starting point for the proposed network architecture. High-throughput computing blocks utilizing weight-stationary systolic networks and accelerated filtering operations have successfully realized low-delay modular polynomial reduction for schemes like RLWE and Saber [15]. Additionally, systolic-based hardware engines tailored specifically for BRLWE and Saber have been reported, exploiting hardware-software co-design methodologies to handle diverse parameter configurations while achieving balanced area-time metrics on programmable logic arrays; nevertheless, such implementations commonly emphasize high-throughput servers rather than the edge-tailored serial processing pipelines developed in this study [13].

Regarding the Saber protocol, dedicated hardware accelerators and polynomial processing cores utilize short secret vectors combined with parallel schoolbook multiplication schemes to achieve full KEM execution while allowing flexible trade-offs between performance and layout area [16]. Later architectural

improvements introduce consolidated coefficient multiplication engines and DSP-centric data packing methods to further lower flip-flop and memory utilization while maintaining high processing speeds [17].

Academic explorations into BRLWE architectures have focused heavily on fault-tolerant, low-complexity computing cells, featuring hybrid-field multipliers that exploit binary-integer properties to compress logic element counts and register storage. These initiatives have produced BRLWE-dedicated cryptoprocessors optimized for peripheral network nodes, alongside lightweight, secure designs engineered to withstand side-channel analysis [11,18]. Moreover, past works have presented inverted BRLWE hardware cores in both high-velocity and ultra-compact frameworks, validating that lattice-driven security protocols can conform to stringent energy and area constraints at the network edge [2].

Specialized processing frameworks utilizing bit-serial routing pipelines and linear-feedback shift registers (LFSRs) deliver supplementary area-time trade-off enhancements for dedicated BRLWE procedures. Certain designs also utilize lookup-table-based finite-field arithmetic to enhance efficiency. However, many current methodologies rely on parallel coefficient loading or target generalized inverted BRLWE applications without fully leveraging systolic regularity [14,19]. Consequently, this work presents a systolic array specifically engineered for polynomial multiplication in BRLWE-based schemes. Unlike general-purpose multipliers, this architecture is designed as a specialized computational primitive for the IoT edge. It merges the consistent throughput of a systolic organization with the area efficiency necessary for embedded nodes, offering a targeted and scalable building block for quantum-resilient security in resource-constrained hardware security modules (HSMs).

The structural blueprint of this manuscript is arranged as follows. First, Section 2 provides the mathematical groundwork and formal constraints governing polynomial multiplication over hybrid ring systems. Advancing from this foundation, Section 3 analyzes the construction of the corresponding algorithmic dependence graphs and explains the sequential mapping steps applied to them. Next, Section 4 delivers a detailed breakdown of the developed systolic array framework, detailing the internal multiplier logic, the behavioral synthesis flow, and critical hardware-level optimization methodologies. Subsequently, Section 5 evaluates the computational efficiency of the processor through comprehensive silicon area and timing evaluations, comparing the performance metrics against state-of-the-art designs. Lastly, Section 6 outlines the core conclusions of this study and summarizes the key technological contributions achieved.

2 Polynomial Multiplication over Hybrid Algebraic Structures

This research investigates the arithmetic evaluation involved in computing the product of two distinct polynomials, $E(y)$ and $F(y)$, defined over a structured algebraic ring environment. The polynomial operand $E(y)$ is parameterized by coefficients mapped from the integer ring $\mathbb{W}_\beta$ modulo β. Conversely, the coefficient elements of the secondary polynomial operand $F(y)$ are strictly restricted to binary quantities $\{0,1\}$. The maximum degree limits for both algebraic polynomials are strictly constrained by the upper positive integer boundary u.

$$E(y) = \sum_{j=0}^{u-1} e_j y^j, \quad \text{where } e_j \in \mathbb{W}_\beta, \tag{1}$$

$$F(y) = \sum_{i=0}^{u-1} f_i y^i, \quad \text{where } f_i \in \{0,1\}. \tag{2}$$

The fundamental operation involves calculating the product $T(y) = E(y) \cdot F(y)$ under the dual constraints imposed by the ring $Q_\beta = \mathbb{W}_\beta[y]/(y^u + 1)$. This requires performing polynomial reduction modulo $y^u + 1$ alongside integer coefficient reduction modulo β. As established in [15], applying the schoolbook multiplication algorithm leads to the following expanded expression:

$$T(y) = \sum_{i=0}^{u-1}\sum_{j=0}^{u-1} f_i e_j \cdot y^{i+j} \bmod (y^u + 1, \beta)$$
$$= \sum_{i=0}^{u-1}\left(\sum_{j=0}^{u-1}(-1)^{\lfloor (i+j)/u \rfloor} \cdot f_i e_j \bmod \beta\right) \cdot y^{(i+j) \bmod u} \tag{3}$$

The ultimate coefficients comprising the resulting product polynomial $T(y)$ are entirely bounded within the ring space $\mathbb{W}_\beta$.

An important architectural benefit of this mathematical environment is that the modulus parameter β is not strictly required to be a prime value. Choosing β to be an exact power of two, such as $\beta = 2^v$, considerably simplifies the otherwise resource-heavy coefficient modular reduction process. This design decision completely replaces intricate, general-purpose residue reduction routines with an extremely hardware-friendly bitwise truncation procedure. Under this approach, modular evaluation of any transient variable is achieved effortlessly by isolating and preserving its v lower-order bits, with the bit-width calculated as $v = \lceil \log_2(\beta) \rceil$. This structural simplification opens up opportunities for substantial software and hardware-level refinements. By completely bypassing the need for dedicated modular division or reduction networks—while simultaneously minimizing the required data word length—vital silicon area and processing cycles are preserved. These saved hardware resources can be systematically utilized to upgrade the core multiplier array, such as implementing dense parallel computing paths. Consequently, this optimization directly accelerates processing throughput and elevates the global performance metrics of the polynomial multiplication engine.

3 Extracting Dependency Graph

To investigate the internal structural dataflow interactions inside the modular polynomial multiplier, an illustrative execution scenario choosing the parameter value $u = 5$ is analyzed. The underlying arithmetic processing is governed by the following operational expression:

$$T(y) = E(y) \cdot F(y) \bmod (y^5 + 1, \beta), \tag{4}$$

yielding a resultant polynomial with a degree strictly less than five:

$$T(y) = t_0 + t_1 y + t_2 y^2 + t_3 y^3 + t_4 y^4. \tag{5}$$

The input polynomials are specified as follows:

$$E(y) = e_0 + e_1 y + e_2 y^2 + e_3 y^3 + e_4 y^4, \tag{6}$$
$$F(y) = f_0 + f_1 y + f_2 y^2 + f_3 y^3 + f_4 y^4. \tag{7}$$

Performing the direct multiplication of $E(y)$ and $F(y)$ produces an initial, unreduced product polynomial of degree eight:

$$T'(y) = t'_0 + t'_1 y + t'_2 y^2 + t'_3 y^3 + t'_4 y^4 + t'_5 y^5 + t'_6 y^6 + t'_7 y^7 + t'_8 y^8. \tag{8}$$

The reduction process employs the polynomial identity $y^5 \equiv -1$, leading to a series of substitutions for higher-order terms: $y^6 \equiv -y$, $y^7 \equiv -y^2$, and $y^8 \equiv -y^3$. By systematically applying these identities and collecting like terms, the final set of reduced coefficients is derived as follows:

$$
\begin{aligned}
t_4 &= e_4 f_0 + e_3 f_1 + e_2 f_2 + e_1 f_3 + e_0 f_4, \\
t_3 &= e_3 f_0 + e_2 f_1 + e_1 f_2 + e_0 f_3 - e_4 f_4, \\
t_2 &= e_2 f_0 + e_1 f_1 + e_0 f_2 - e_4 f_3 - e_3 f_4, \\
t_1 &= e_1 f_0 + e_0 f_1 - e_4 f_2 - e_3 f_3 - e_2 f_4, \\
t_0 &= e_0 f_0 - e_4 f_1 - e_3 f_2 - e_2 f_3 - e_1 f_4.
\end{aligned} \tag{9}
$$

The expression given in Eq. (9) can be generalized into a unified formulation that accounts for the modular reduction $y^u \equiv -1$.

$$
t_k = \sum_{i=0}^{u-1} (-1)^{\lfloor (i+(u-1-j))/u \rfloor} e_{u-1-j} f_i \pmod{\beta}; \quad (u-1-j) = (k-i) \pmod{u} \tag{10}
$$

In this representation, where $k, i, j \in \{0, \ldots, u-1\}$, the term $\lfloor (i + (u-1-j))/u \rfloor$ serves as the primary mechanism for determining whether partial products are added to or subtracted from the accumulating result. Since the indices i and j are strictly bounded by the polynomial degree and the array dimensions, the value of this floor function is binary, evaluating to either 0 or 1 and acting as a built-in decision operator for the hardware; specifically, when $(i + (u-1-j)) < u$, the term $\lfloor (i + (u-1-j))/u \rfloor = 0$, resulting in a positive sign ($(-1)^0 = 1$) for addition mode, whereas when $(i + (u-1-j)) \geq u$, the term $\lfloor (i + (u-1-j))/u \rfloor = 1$, resulting in a negative sign ($(-1)^1 = -1$) for subtraction mode.

To visually map out the architectural constraints mathematically defined in Eq. (10), the algorithm is projected onto the dependence graph (DG) illustrated in Fig. 1. This graphical representation explicitly outlines how the operational tasks relate to one another within the modular multiplication framework. An examination of the DG shows a methodical dataflow pattern: the coefficients e_{u-1-j} are allocated vertically and maintain a fixed assignment to specific nodes, while the coefficients f_i are broadcast horizontally across each row. By maintaining e_{u-1-j} at fixed nodes and propagating the intermediate partial results t_k diagonally, the architecture effectively realizes the $k = (i + (u-1-j)) \pmod{u}$ relationship. This ensures that the partial products are accumulated into the correct result coefficient t_k as they move through the array, while the term $\lfloor (i + (u-1-j))/u \rfloor$ ensures the proper sign is applied according to the reduction rule. A critical architectural aspect is the feedback of the output partial result t from the nodes in the rightmost column to those in the leftmost column, as illustrated in Fig. 1. The resulting coefficients of the polynomial product are ultimately gathered from the output nodes located at the bottom of the DG structure.

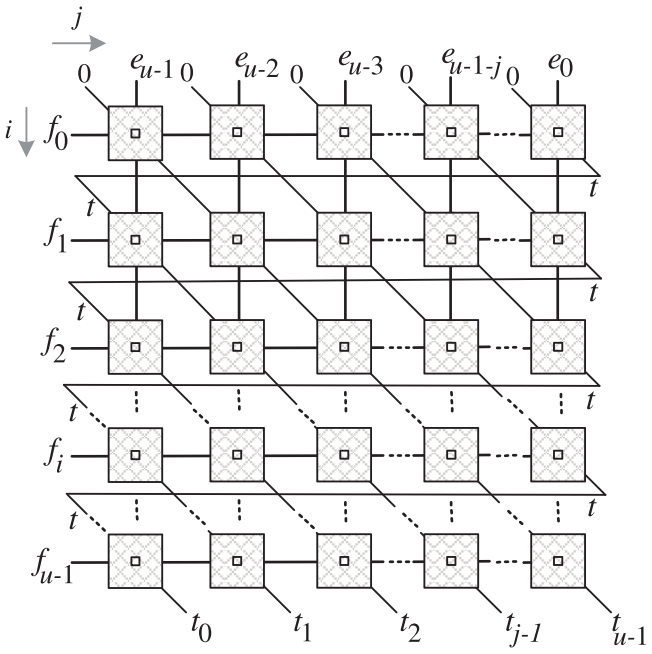

Figure 1: Computational DG for the modular multiplier.

4 Development of the Systolic Array Architecture

To systematically derive the systolic array architecture, the scheduling and projection vectors for the DG must be precisely determined. Unlike conventional ad-hoc methods that focus on a single hardware instance, the methodology used here establishes a systematic design-space exploration framework. Adopting established methodologies from prior work [20,21], the scheduling vector **H** and projection vector **V** are formulated to guide the systolic array's design [22,23]. This parameterized approach allows for the mathematical navigation of various architectural configurations, enabling the selection of an optimally balanced design tailored to specific application constraints, such as area or throughput.

As outlined in [20,21,23], the scheduling vector is obtained from a scheduling function $\mathbf{G}(\mathbf{K})$ defined as:

$$\mathbf{G}(\mathbf{K}) = \mathbf{HK} - b = ih_0 + jh_1 - b \tag{11}$$

In this formulation, $\mathbf{K} = [i\ j]$ denotes the positional vector of a node within the DG. The components h_0 and h_1 constitute the scheduling vector $\mathbf{H} = [h_0\ h_1]$. To prevent the assignment of negative timesteps to any node in the DG, a constant b is incorporated into the scheduling equation. For the present design, a value of $b = 1$ is selected, which ensures that all nodes are scheduled at positive time instances.

Through analysis of the inter-node dependencies in the DG, the scheduled execution time $\mathbf{G}(\mathbf{K})$ for each node can be established. This scheduling directly determines the components h_0 and h_1 of the vector **H** according to Eq. (11). Although multiple valid scheduling vectors satisfy Eq. (11), the selection $\mathbf{H} = [1\ 0]$ produces the most efficient systolic architecture.

Furthermore, as established in [20,21], the projection vector **V** and scheduling vector **H** must satisfy the following constraint:

$$\mathbf{HV} \neq \mathbf{0} \tag{12}$$

Given the scheduling vector under the choice of $\mathbf{H} = [1\ 0]$ alongside the geometric boundaries outlined in Eq. (12), multiple candidate projection vectors remain valid. Among these available options, the specific projection vector $\mathbf{V} = [1\ \ -1]$ is opted for, since it yields a physically constructible and highly effective systolic array.

The spatial allocation mapping a discrete dependency graph node $\mathbf{K}$ onto an explicit processing element (PE) inside the systolic array is governed by the following projection function $\mathbf{X}$:

$$\mathbf{X}(\mathbf{K}) = \mathbf{C}\,\mathbf{K} \tag{13}$$

where the symbol $\mathbf{C}$ represents the projection matrix. Consistent with parallel computing synthesis principles [21], the chosen projection vector $\mathbf{V}$ must occupy the null space spanned by this specific matrix. Consequently, matching the projection vector $\mathbf{V} = [1\ \ -1]$ defines the exact linear operator as $\mathbf{C} = [1\ 1]$.

Evaluating every individual dependency graph node $\mathbf{K}(i, j)$ using this projection matrix $\mathbf{C} = [1\ \ 1]$ in tandem with the scheduling vector $\mathbf{H} = [1\ \ 0]$ yields the projection function $\mathbf{X}(\mathbf{K})$ and the scheduling function $\mathbf{G}(\mathbf{K})$ formulated below. These functions are vital for organizing the execution timeline of individual nodes and properly distributing them to their designated PE across the computing network [20,21]. The deliberate pairing of these scheduling and projection mechanisms guarantees synchronous coordination and optimal throughput performance throughout the physical systolic framework [22,23].

$$\mathbf{G}(\mathbf{K}) = i;\ \ \mathbf{X}(\mathbf{K}) = i + j \tag{14}$$

Subjecting the algorithmic graph to temporal mapping via the scheduling function $\mathbf{G}(\mathbf{K})$ yields the specific clock cycle distribution illustrated in Fig. 2. Following this timing assignment, the projection function $\mathbf{X}(\mathbf{K})$ transforms the spatial coordinates, routing the dependency nodes onto the physical systolic hardware layout diagrammed in Fig. 3. This synthesized systolic array consists of exactly u computing blocks (PEs)—instantiated here with a size of $u = 5$—and concludes the entire algebraic operation within a span of u sequential clock cycles, ultimately generating the complete set of target coefficients t via a parallel output interface.

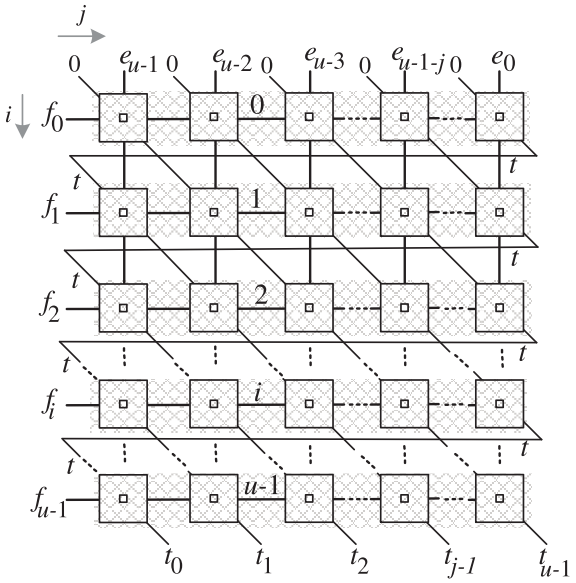

Figure 2: Node scheduling distribution.

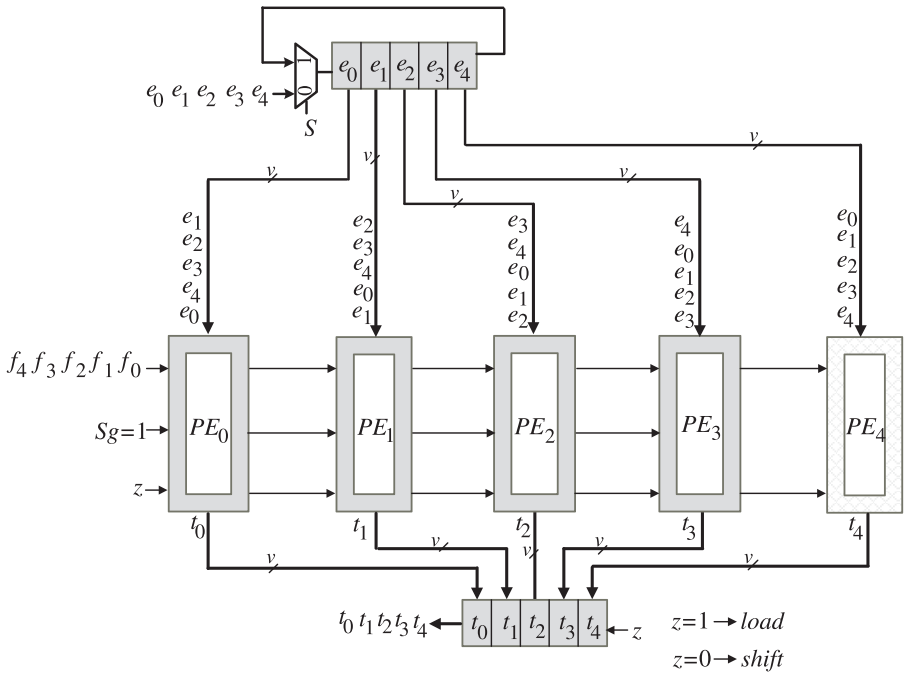

Figure 3: Proposed systolic array architecture for polynomial multiplication ($u = 5$).

To manage the serial input of the coefficients e_0, e_1, e_2, e_3, e_4 for polynomial E, a rotate left shift register (RLSH) is integrated at the top of the systolic array, as shown in Fig. 3. This register cyclically shifts the coefficients, supplying the appropriate e value in parallel to each processing element on every clock cycle. Conversely, to serialize the output coefficients t_0, t_1, t_2, t_3, t_4 of the product polynomial T, a parallel-to-serial left shift register (PSLSR) is placed at the array's output.

A pipelined control signal, Sg, is propagated between the general processing elements (PE_{j-1}) to dynamically configure their arithmetic logic for either addition or subtraction. Figs. 4 and 5 provide a detailed view of the inner workings of the general PE (PE_{j-1}) and the terminal PE (PE_{u-1}). The internal logic of the proposed general PE (PE_{j-1}) is specifically optimized to exploit the binary nature of the BRLWE scheme. Since the coefficients $f_i \in \{0, 1\}$, the partial product generation is simplified from a complex multi-bit multiplier to a parallel bank of v two-input AND gates. This structural simplification significantly reduces the critical path and silicon area, contributing to the high operating frequency of the design.

The general PE contains a primary datapath featuring the AND-based multiplier, a v-bit ripple-carry adder, and a series of v two-input XOR gates. Control and data flow are managed by a 1-bit Flip-Flop (D_s) and a v-bit Flip-Flop array (D_t). The modular reduction is handled natively by mapping the previously discussed floor function logic to the Sg control signal, allowing the architecture to implement the required subtraction ($T - G$) without additional stages. To achieve this, the Sg signal facilitates 2's complement arithmetic ($T + \bar{G} + 1$) through a dual-purpose toggle. When $Sg = 1$, it triggers a bit-wise inversion of the product via the XOR gates and simultaneously provides the required "+1" by serving as the carry-in (C_{in}) for the v-bit adder. Such a structural arrangement enables the PE to switch dynamically between subtraction and addition operations within a single clock period, preserving the uniform execution pattern of the systolic array without introducing any additional logic delay or area penalty. Throughout this framework, each incoming operand coefficient e represents a v-bit signed value, formatted in two's complement representation as $(e_0 e_1 \ldots e_{v-1})$.

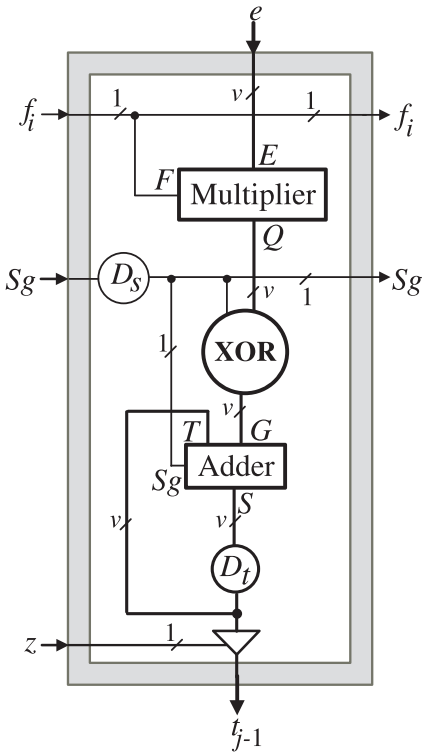

Figure 4: General processing element (PE_{j-1}) logic schematic.

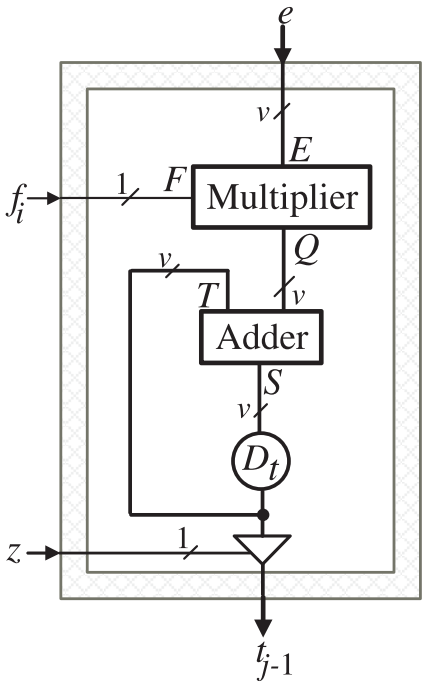

Figure 5: Internal logic architecture of the terminal processing element (PE_{u-1}).

Conversely, the terminal PE (PE_{u-1}), illustrated in Fig. 5, features a streamlined architectural layout. This specific cell incorporates a v-bit multiplication block, a v-bit addition unit, and a set of v-bit registers

(D_t). While the internal multiplier within this boundary element relies on a matrix of v dual-input AND gates identical to the standard PE configuration, its corresponding adder is constructed as a ripple-carry topology utilizing basic half-adder (HA) structures.

A precise initialization sequence is required to ensure correct operation of the systolic architecture before computation begins. All Flip-Flops must first be reset to a zero state. The multiplexer control signal in Fig. 3 must be set to $S = 0$ to enable loading the coefficients e_0, e_1, e_2, e_3, e_4 into the shift register.

The signal Sg controls the arithmetic mode of processing elements PE_0 through PE_3, configuring them for addition when $Sg = 0$ and subtraction when $Sg = 1$. Simultaneously, the control signal z must be set to 0 to disable the tri-state buffers, thus isolating the systolic array outputs from the parallel-to-serial shift register (PSLSR) inputs. This initialization phase is crucial for establishing deterministic operation throughout subsequent computational stages.

The multiplier's performance stems from a carefully engineered control sequence that balances computational throughput with numerical precision. The architecture employs general processing elements (PE_{j-1}) with dual-mode capability, dynamically switching between addition and subtraction operations. A globally distributed control signal Sg synchronizes these mode transitions across the processing pipeline, maintaining proper functional alignment. The subsequent analysis examines this operational sequence in detail.

1. **Initialization Phase:**
   - At clock cycle $i = 0$, the coefficients $e_0, e_1, \cdots, e_{u-1}$ from polynomial E are transferred into their designated processing elements (PEs). Simultaneously, the initial coefficient bit f_0 of polynomial F is distributed across all PE units, as illustrated in Figs. 3–5. A global reset signal clears the D_t Flip-Flops in every PE, which resets all intermediate partial results (t) to zero and sets the starting state for the computation.
   - Simultaneously, the D_s Flip-Flops are cleared, forcing the XOR gate inputs and the adder's carry-in signals associated with the Sg line to a zero state. This configuration ensures that every PE is initialized to function in addition mode.
   - The control signal z is subsequently transitioned to a low state ($z = 0$), which turns off the tri-state buffers located at the PE outputs and separates the systolic array from the inputs of the parallel-serial-load shift register ($PSLSR$).
2. **Computation Phase:**
   - During clock cycle $i = 1$, each PE accumulates the intermediate results t generated in the preceding cycle. The coefficient bit f_1 is distributed to the inputs of all PEs simultaneously. The control signal Sg is set to high ($Sg = 1$) and pipelined specifically to PE_0, which triggers a configuration for subtraction mode, while the remaining PEs continue to operate in addition mode.
   - In clock cycle $i = 2$, each PE accumulates the newly updated intermediate results t. The coefficient bit f_2 is simultaneously broadcast to all PE units. As the control signal Sg propagates to PE_1, it shifts that unit into subtraction mode; consequently, both PE_0 and PE_1 now function in subtraction mode while the rest of the PEs persist in addition mode.
   - Broadly, during clock cycle i, each PE accumulates the intermediate values t derived from the previous cycle $i - 1$. All PE units receive the broadcasted coefficient bit f_i simultaneously. At this stage, the control signal Sg advances to PE_{i-1}, transitioning it into subtraction mode. As a result, the units PE_0 through PE_{i-1} function in subtraction mode, whereas the rest of the PEs stay in addition mode.

- During the concluding computation cycle ($i = u - 1$), each PE accumulates the intermediate results t obtained from cycle $u - 2$. The final coefficient bit f_{u-1} is broadcast across all PE units simultaneously. As the control signal Sg arrives at PE_{u-2}, it transitions that unit into subtraction mode. Consequently, the range of processing elements from PE_0 to PE_{u-2} functions in subtraction mode, while the terminal unit (PE_{u-1}) continues to operate in addition mode.

3. **Output Phase:**
   - Upon the conclusion of cycle $u - 1$, the control signal z is set to high ($z = 1$), which engages the tri-state buffers. This action allows the final coefficients t to be discharged in a parallel fashion, signaling the successful termination of the multiplication process.
   - To facilitate serial output, the terminal coefficients t are transferred into the parallel-to-serial left shift register ($PSLSR$) by setting z to high ($z = 1$) at the conclusion of cycle $u - 1$. Once the loading phase is finished, z is returned to a low state ($z = 0$) to permit the serial shifting process. Consequently, the coefficients are provided sequentially across the following u clock cycles.

5 Findings and Analysis

This evaluates the proposed hardware architecture from both theoretical and implementation perspectives. First, a detailed complexity analysis is presented, examining the underlying structure and resource requirements of the polynomial multiplication unit. Subsequently, an empirical performance assessment compares the design against contemporary implementations, highlighting its efficiency in terms of area, speed, and overall hardware economy.

5.1 Complexity Insights

This section analyzes the algorithmic complexity of the proposed hardware design for computing the product of two polynomials: $E(y)$, which has coefficients from the modular integer ring $\mathbb{W}_\beta$, and $F(y)$, characterized by binary coefficients. Both polynomials are constrained by the parameter u in terms of degree.

At the core of the computation, the design employs u independent multipliers, each processing v-bit operands. Collectively, these multipliers require vu two-input AND gates to enable parallelized multiplication, thus enhancing operational throughput. To amalgamate the intermediate results, the architecture integrates $v(u-1)$ two-input XOR gates and u ripple-carry adders, each handling v-bit values. These adder modules are constructed from $2vu$ AND gates, vu OR gates, and $2vu$ XOR gates, ensuring accurate arithmetic summation.

Data storage is managed through vu flip-flops for coefficient pipelining corresponding to the partial product u, supplemented by $u - 1$ additional flip-flops for controlling the mode-selection signal Sg.

The overall hardware footprint can be approximated as follows:

- Tri-state buffers: vu
- AND gates: $4vu - 2v$
- OR gates: uv
- XOR gates: $3uv - 2v$
- Flip-flops (FFs): $3uv + u - 1$

This enumeration includes storage units located within the input and output shift registers of the systolic array, providing a comprehensive account of the sequential logic involved.

An important operational feature arises from connecting the carry input of the adder to the signal Sg. This connection allows the same hardware to perform both addition and subtraction in 2's complement arithmetic, contingent on the value of Sg as previously defined.

With respect to timing performance, the critical path delay is identified as $T_A + 3T_X$, where T_A represents the delay of a two-input AND gate and T_X corresponds to the delay of a two-input XOR gate. This specific path consists of the AND gate used for partial product generation, followed by three XOR gates: one serving as the conditional inverter for the Sg control signal and two residing within the full-adder logic of the v-bit ripple-carry structure. The calculation of the complete set of coefficients for $T(y)$ is finalized in u clock cycles, which highlights the architecture's ability to perform high-speed polynomial multiplication within ring-based parameters. This streamlined path ensures that the hardware can maintain high operating frequencies while the harmonious operation of all constituent parts ensures effective and consistent performance in practical use cases.

5.2 Implementation Overview and Performance Insights

A comprehensive performance assessment of the introduced architectural approach was conducted by benchmarking against modern implementations documented in the literature [13–17]. To demonstrate the scalability and efficiency of the proposed architecture, we evaluated the design using $(u, v) = (256, 8)$ and $(u, v) = (512, 8)$, as these configurations represent critical benchmarks for both specialized and standardized cryptographic systems. Specifically, the $u = 256$ dimension was selected to align with the polynomial dimensions widely adopted by state-of-the-art NIST-standardized lattice-based cryptographic schemes, while maintaining a $v = 8$ bit-width to ensure high precision and energy efficiency for IoT applications. By achieving high performance at these dimensions, the results confirm that the proposed systolic approach is not only optimized for low-power BRLWE but is also architecturally compatible with the fundamental parameters required for contemporary post-quantum security protocols.

This comparative study evaluates the physical FPGA implementation results for the $(u, v) = (256, 8)$ and $(u, v) = (512, 8)$ configurations. The proposed design, along with the referenced architectures, was implemented in VHDL, synthesized for the Xilinx UltraScale+ platform using Vivado 2019.2, and verified through simulation with ModelSim prior to physical implementation.

Table 1 summarizes the implementation characteristics obtained, presenting quantitative measures of both resource allocation and temporal behavior. The metrics included quantify the usage of configurable logic resources specifically LUTs (Look-Up Tables) and FFs, the maximum sustainable clock frequency (Fmax), and the computational latency in cycles. From these foundational measurements, two aggregate performance indicators are computed: overall processing time, determined by multiplying critical path delay by cycle count (Latency), and the area-delay product (ADP), formed as the product of LUT consumption and total delay. The ADP offers a consolidated metric that reflects the architectural balance between hardware investment and operational throughput.

The data compiled in Table 1 facilitates a detailed comparison between the new design and prior architectures across both parameter configurations. The proposed system consistently demonstrates an optimized equilibrium between hardware requirements and processing capability, as evidenced by improved ADP figures. To ensure equivalence in comparison, input and output shift registers associated with the systolic data path were excluded from the reported resource totals, thereby maintaining consistency with the accounting methodology employed in the compared implementations.

The proposed design exhibits significant improvements across all key metrics compared to existing designs [13–17], showcasing clear advantages in both resource efficiency and operational performance. For the smaller configuration ($u = 256$), the design achieves a remarkably low LUT count of 5508 and a FF count of 2303, indicating substantial reductions over prior work. When compared to the most efficient prior design [14], this corresponds to a 44.0% reduction in LUTs and a 28.7% reduction in FFs. In contrast to the larger design [13], the savings are even more pronounced, showing reductions of 77.3% in LUTs and 55.9%

in FFs. This significant reduction in resource utilization directly contributes to the design's superior ADP. The design achieves an ADP of 3740, which is less than one-fourth of the next best design [17], which has an ADP of 15,656.

Table 1: FPGA area-time complexity comparison for for $u = 256$ and $u = 512$.

Design	u	LUT	FF	Fmax	Latency	Delay	ADP	% ADP
[13] ($u = 4$)	256	24,248	5220	91	68	0.75	18,119	79.4
[14]	256	9840	3230	133	258	1.95	19,145	80.5
[15]	256	16,564	8755	173	511	2.96	48,954	92.4
[16]	256	17,003	5083	173	256	1.48	25,175	85.1
[17] (HS-I 256)	256	10,574	5150	173	256	1.48	15,656	76.1
Proposed	256	5508	2303	377	256	0.68	3740	–
[13] ($u = 4$)	512	53,345	11,484	72	132	1.85	98,484	79.3
[14]	512	21,385	7106	109	514	4.71	100,659	79.8
[15]	512	36,440	19,262	142	1021	7.20	262,566	92.3
[16]	512	37,407	11,160	140	512	3.65	136,412	85.1
[17] (HS-I 256)	512	23,263	11,330	144	512	3.55	82,542	75.4
Proposed	512	12,118	4607	306	512	1.68	20,308	–

The performance advantages extend beyond resource efficiency to encompass critical operational characteristics. The proposed design achieves an impressive maximum frequency (Fmax) of 377 MHz for $u = 256$, more than doubling the performance of comparable designs, representing a 118% improvement over the 173 MHz achieved by designs in [15–17]. This frequency advantage translates directly into reduced operational delay, with the proposed design completing operations in just 0.68 µs, compared to the range of 0.75 to 2.96 µs for reference designs. The combination of reduced area and improved speed yields an outstanding ADP reduction ranging from 76.1% to 92.4% compared to existing approaches, thereby validating the architectural efficiency of the proposed solution.

For the larger configuration ($u = 512$), similar advantages are maintained. The proposed design utilizes 12,118 LUTs and 4607 FFs, achieving consistent percentage reductions across all comparative designs. Most notably, it achieves an operating frequency of 305.5 MHz more than twice the speed of the design reported in [13]; thus maintaining a significant advantage over all alternatives. The operational delay of 1.676 µs is approximately half that of the nearest competitor [17], which reports a delay of 3.548 µs. These performance gains culminate in an ADP of 20,308, reflecting reductions of 75.4% to 92.3% compared to existing designs, with particularly strong advantages over the resource-intensive approaches outlined in [15,16].

The experimental outcomes collectively underscore that the proposed hardware architecture adopts a holistic optimization strategy, systematically harmonizing area efficiency with computational throughput. Rather than focusing exclusively on a single performance dimension, the design achieves a carefully calibrated balance that avoids both excessive hardware overhead for marginal speed improvements and detrimental performance compromises in the pursuit of footprint reduction. This equilibrium results in consistently superior ADP metrics across all evaluated configurations, establishing a Pareto-optimal trade-off that is highly effective for FPGA implementations.

These characteristics make the architecture particularly well-suited for deployment in resource-constrained IoT edge modules, where stringent constraints on cost and physical size coexist with demands for responsive, real-time processing. The system's robust performance scalability across multiple operational

parameters further demonstrates its architectural resilience and enhances its viability for diverse real-world embedded applications requiring efficient, serial-data processing.

Notably, the potential of the proposed design as a high-performance computational primitive within the BRLWE post-quantum cryptographic framework stands out. While full system-level integration remains a future milestone, embedding this specialized multiplier core into such a lattice-based cryptographic system is anticipated to significantly improve operational efficiency, making it a targeted choice for securing next-generation IoT edge nodes. This enhancement provides the architectural foundation necessary to fortify cryptographic resilience against emerging quantum computing threats, reinforcing the overall trustworthiness of IoT infrastructures within an increasingly complex cybersecurity landscape. By directly addressing urgent security vulnerabilities posed by advanced adversarial capabilities, the architecture fosters greater confidence in IoT ecosystems and encourages broader adoption across industrial, commercial, and consumer domains. Its alignment with SDG 8 (Decent Work and Economic Growth) and SDG 9 (Industry, Innovation, and Infrastructure) is particularly notable. By enabling secure, efficient, and scalable edge computation, it supports the advancement of innovation-driven, resilient economic activities within an increasingly digital global economy.

Ultimately, the synergy between fortified post-quantum security and optimized hardware performance delineates a critical pathway for the evolution of trusted IoT systems. Thus, the architecture represents not merely an incremental improvement but a foundational step toward sustainable and secure digital transformation.

6 Overview and Conclusion

The swift proliferation of IoT deployments demands forward-looking defense mechanisms resilient against impending quantum breakthroughs. This research addresses this vulnerability by optimizing polynomial multiplication—the central processing bottleneck within the Binary Ring Learning With Errors (BRLWE) protocol. The hardware processor developed functions as a dedicated, high-throughput arithmetic primitive, mathematically streamlined for resource-limited IoT peripheral components. Exploiting a structured index transformation methodology alongside embedded modular reduction pathways allows the layout to attain heightened clock frequencies while compressing its overall silicon area. Mitigating the consumption of logic resources—specifically LUTs and edge-triggered registers—yields an optimized building block that safeguards distributed nodes and encourages secure electronic commerce. Furthermore, by reinforcing the cryptographic defenses of networked devices, this contribution directly satisfies UN Sustainable Development Goals (SDGs 8 and 9) for adaptive industrial infrastructure and economic development. This framework propels the practical utility of post-quantum cryptography, supporting the migration toward verified, secure IoT ecosystems. Although this contribution centers on the underlying multiplication circuit, it establishes a scalable design framework compatible with forthcoming NIST post-quantum suites. Subsequent efforts will focus on embedding this block within a unified, top-level BRLWE cryptographic processor to benchmark system-level execution metrics, alongside executing extensive side-channel power and timing vulnerability evaluations to guarantee structural resilience in hostile deployments.

Acknowledgement: The authors extend their appreciation to Prince Sattam bin Abdulaziz University for funding this research work through the project number (PSAU/2025/01/34935).

Funding Statement: This research was funded by Prince Sattam bin Abdulaziz University, grant number PSAU/2025/01/34935.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Atef Ibrahim; methodology, Atef Ibrahim and Fayez Gebali; software, Atef Ibrahim; validation, Atef Ibrahim; formal analysis, Atef

Ibrahim; investigation, Atef Ibrahim; resources, Atef Ibrahim; data curation, Atef Ibrahim; writing—original draft preparation, Atef Ibrahim; writing—review and editing, Atef Ibrahim and Fayez Gebali; visualization, Atef Ibrahim; supervision, Atef Ibrahim; project administration, Atef Ibrahim; funding acquisition, Atef Ibrahim All authors have read and agreed to the published version of the manuscript. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Diop M, Ndiaye P, Dione D, Diop I. IoT security in the quantum era: state of the art and open challenges. In: Proceedings of the 2025 5th International Conference on Innovative Research in Applied Science; 2025 May 15–16; Fez, Morocco. New York, NY, USA: Engineering and Technology (IRASET); 2025. p. 1–11.
2. Ebrahimi S, Bayat-Sarmadi S, Mosanaei-Boorani H. Post-quantum cryptoprocessors optimized for edge and resource-constrained devices in IoT. IEEE Internet Things J. 2019;6(3):5500–7.
3. Rathod P, Bhatt N, Nema R, Jyotheeswari P. A comprehensive review on IoT security challenges and solutions. In: Proceedings of the 2025 Seventh International Conference on Computational Intelligence andCommunication Technologies (CCICT); 2025 Apr 11–12; Sonepat, India. New York, NY, USA: IEEE; 2025. p. 696–701.
4. Regev O. On lattices, learning with errors, random linear codes, and cryptography. J ACM. 2009;56(6):1–40.
5. Buchmann J, Göpfert F, Player R, Wunderer T. On the hardness of LWE with binary error: revisiting the hybrid lattice-reduction and meet-in-the-middle attack. In: Proceedings of the International Conference on Cryptology in Africa; 2016 Apr 13–15; Fes, Morocco. Berlin/Heidelberg, Germany: Springer; 2016. p. 24–43.
6. Asif R. Post-quantum cryptosystems for Internet-of-Things: a survey on lattice-based algorithms. IoT. 2021;2(1):71–91. doi:10.3390/iot2010005.
7. Dam DT, Tran TH, Hoang VP, Pham CK, Hoang TT. A survey of post-quantum cryptography: start of a new race. Cryptography. 2023;7(3):40. doi:10.3390/cryptography7030040.
8. Bos J, Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schanck JM, et al. CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM. In: Proceedings of the 2018 IEEE European Symposium on Security and Privacy (EuroS&P); 2018 Apr 26–28; London, UK. New York, NY, USA: IEEE; 2018. p. 353–67.
9. D'Anvers JP, Karmakar A, Sinha Roy S, Vercauteren F. Saber: module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM. In: Proceedings of the International Conference on Cryptology in Africa; 2018 May 7–9; Marrakesh, Morocco. Berlin/Heidelberg, Germany: Springer; 2018. p. 282–305.
10. Buchmann J, Göpfert F, Güneysu T, Oder T, Pöppelmann T. High-performance and lightweight lattice-based public-key encryption. In: Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security; 2016 May 30; Xi'an, China. p. 2–9.
11. He P, Guin U, Xie J. Novel low-complexity polynomial multiplication over hybrid fields for efficient implementation of binary ring-LWE post-quantum cryptography. IEEE J Emerg Sel Top Circuits Syst. 2021;11(2):383–94. doi:10.1109/jetcas.2021.3075456.
12. Basso A, Bos JW, D'Anvers JP, Karmakar A, Mera JMB, Renes J, et al. Using learning with rounding to instantiate post-quantum cryptographic algorithms. Cryptol EPrint Arch. 2025. doi:10.1145/3807514.
13. Bao T, He P, Xie J. Systolic acceleration of polynomial multiplication for KEM saber and binary ring-LWE post-quantum cryptography. In: Proceedings of the 2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST); 2022 Jun 27–30; McLean, VA, USA. New York, NY, USA: IEEE; 2022. p. 157–60.
14. Xie J, He P, Wen W. Efficient implementation of finite field arithmetic for binary ring-LWE post-quantum cryptography through a novel lookup-table-like method. In: Proceedings of the 2021 58th ACM/IEEE Design

Automation Conference (DAC); 2021 Dec 5–9; San Francisco, CA, USA. New York, NY, USA: IEEE; 2021. p. 1279–84.
15. Tan W, Wang A, Zhang X, Lao Y, Parhi KK. High-speed VLSI architectures for modular polynomial multiplication via fast filtering and applications to lattice-based cryptography. IEEE Trans Comput. 2023;72(9):2454–66. doi:10.1109/tc.2023.3251847.
16. Roy SS, Basso A. High-speed instruction-set coprocessor for lattice-based key encapsulation mechanism: saber in hardware. IACR Trans Cryptogr Hardw Embed Syst. 2020;2020(4):443–66. doi:10.46586/tches.v2020.i4.443-466.
17. Basso A, Roy SS. Optimized polynomial multiplier architectures for post-quantum KEM saber. In: Proceedings of the 2021 58th ACM/IEEE Design Automation Conference (DAC); 2021 Dec 5–9; San Francisco, CA, USA. New York, NY, USA: IEEE; 2021. p. 1285–90.
18. Ebrahimi S, Bayat-Sarmadi S. Lightweight and fault-resilient implementations of binary ring-LWE for IoT devices. IEEE Internet Things J. 2020;7(8):6970–8. doi:10.1109/jiot.2020.2979318.
19. Imaña JL, He P, Bao T, Tu Y, Xie J. Efficient hardware arithmetic for inverted binary ring-LWE based post-quantum cryptography. IEEE Trans Circuits Syst I Regul Pap. 2022;69(8):3297–307. doi:10.1109/tcsi.2022.3169471.
20. Ibrahim A. Efficient parallel and serial systolic structures for multiplication and squaring over GF (2^m). Can J Electr Comput Eng. 2019;42(2):114–20. doi:10.1109/cjece.2019.2900087.
21. Gebali F. Algorithms and parallel computing. Hoboken, NJ, USA: John Wiley & Sons; 2011.
22. Ibrahim A. Low-space bit-serial systolic array architecture for interleaved multiplication over GF (2^m). IET Comput Digit Tech. 2021;15(3):223–9.
23. Ibrahim A, Gebali F. Enhancing field multiplication in IoT nodes with limited resources: a low-complexity systolic array solution. Appl Sci. 2024;14(10):4085. doi:10.3390/app14104085.