Computer Networks and Communications
https://ojs.wiserpub.com/index.php/CNC

Research Article

A Novel Approach to Enhance the Security and Efficiency of Binary Ring-LWE for IoT Resource-Constrained

Hadjer Goumidi*, Samuel Pierre

Mobile Computing and Networking Research Laboratory (LARIM), Department of Computer and Software Engineering, Polytechnique Montréal, Montreal, QC, Canada
E-mail: hadjer.goumidi@polymtl.ca

Received: 20 August 2024; **Revised:** 17 December 2024; **Accepted:** 19 December 2024

Abstract: The rapid expansion of the Internet of Things (IoT) brings a vast proliferation of network connections. This surge in connectivity significantly increases the risk of private data exposure during transmission and processing. Traditional public key encryption schemes face considerable challenges due to their high computational complexity and vulnerability to quantum attacks. Recently, Lattice-based cryptography, particularly the Binary Ring Learning With Errors (BRLWE) paradigm, has garnered significant attention for its quantum resistance and lightweight computational requirements. However, BRLWE remains vulnerable to physical attacks, especially Side-Channel Attacks (SCA). This paper proposes a novel 3-Decomposition Karatsuba multiplication-based random shuffling scheme to enhance both the efficiency and security of BRLWE. We evaluate the security performance of our proposed scheme against quantum hybrid attacks and SCAs. We assess the performances of different Karatsuba multiplication techniques in terms of computation cost, energy consumption and memory usage to make choose which Karatsuba technique is suitable for our proposal. Our experimental results show that our proposed approach provides the lowest encryption computation time of 18.97 ms and decryption computation time of 9.53 ms compared to the BRLWE and its improved versions. Furthermore, it improves the security level while it decreases the computation time of the original BRLWE by 32.49% and 20.58%, for the encryption and decryption phases, respectively.

Keywords: internet of things security, post-quantum cryptography, lattice-based cryptography, binary ring learning with errors, shuffling and karatsuba multiplication

1. Introduction

The Internet of Things (IoT) is a well-established area of study that connects the physical world to the Internet through existing network infrastructures. This technology enables smart devices, such as physical devices, vehicles, household appliances, and more, to autonomously collect and share data via the Internet. The 'Cluster of European Research Projects on the Internet of Things' (CERP-IoT) program [1] defines IoT as a dynamic infrastructure within a global network, characterized by self-configuring capabilities based on standardized and interoperable communication protocols. Physical and virtual objects within this network possess unique identities, physical attributes, virtual personalities, and intelligent interfaces, facilitating seamless integration into the network.

The emergence of the IoT concept has led to the development of various applications, such as smart cities, smart industries, smart transportation, smart energy, etc. These applications exhibit specific characteristics, generating vast

DOI: https://doi.org/10.37256/cnc.3120255530

volumes of data while requiring sustained connectivity and energy for extended periods. However, they face challenges related to memory limitations, device capacity, network constraints, and power consumption. Ensuring the security and privacy of the transmitted data within the IoT network is a primordial task. Besides, a robust security measure must take into consideration the limitations of the IoT devices in terms of memory, device capacity and power consumption.

To ensure the secure communication of critical and private data between IoT nodes, it is crucial to implement appropriate encryption primitives. However, conventional encryption schemes like AES [2], RSA [3], and ECC [4] typically involve complex mathematical operations and require significant memory, making them unsuitable for resource-constrained IoT devices. Moreover, classical public key encryption schemes relying on mathematical problems like large integer factorization and discrete logarithms are vulnerable to polynomial-time methods through Shor's quantum algorithms [5, 6]. Simply increasing the key size of existing classical public key encryption schemes will not enhance security against available quantum computers. Unfortunately, the rapid advancement of quantum computing puts at risk the security of these classic encryption schemes, leading to substantial resource overhead. Therefore, there is a pressing need for quantum-resistant and lightweight alternatives to address these challenges effectively. This need has prompted several researchers and organizations to initiate the standardization process of post-quantum cryptography. Figure 1 depicts a taxonomy of different post-quantum cryptography algorithms suggested by different researchers [7, 8, 9, 10].

Original/clear Figures used in our manuscript:

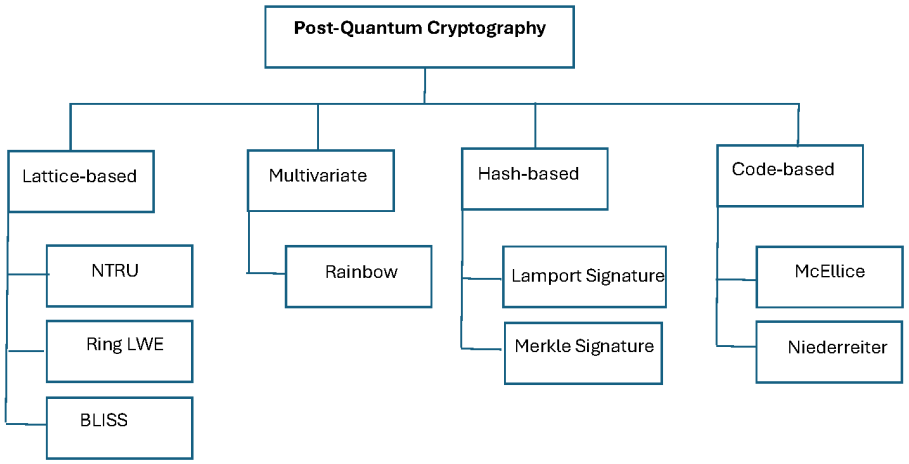

Figure 1. Taxonomy of different post-quantum cryptography algorithms

Among current post-quantum cryptosystems, lattice-based cryptography, particularly the Learning With Errors (LWE) problem [6, 8] and its variants, such as ring learning with error (Ring-LWE) [10] and BRLWE [9], have gained significant attention of researchers. Lattice-based cryptography demonstrates superior efficiency and faster execution times compared to other cryptographic approaches, and unlike code-based cryptography, it does not suffer from large key sizes. Moreover, it provides a robust resistance against attacks leveraging quantum computers, which makes lattice-based public key encryption an ideal choice for resource-constrained IoT nodes [7, 10, 11, 12, 13]. Especially the BRLWE, researchers in [9, 14] proved that the BRLWE is practical for ultralightweight and resource-constrained IoT devices.

Over the last decade, IoT devices have encountered a range of SCA, including timing [15], fault injection attacks [16, 17], Simple Power Analysis (SPA), and Differential Power Analysis (DPA) attacks [12, 14, 18, 19].

Side-channel analysis of BRLWE differs from RLWE because parts of the secret key (i.e., coefficients of the secret-key polynomial) in BRLWE are drawn from a binary distribution, resulting in only two possible values: '0' or '1'. On the other hand, RLWE keys are sampled from a larger set with a Gaussian distribution. Therefore, the evaluation of side-channel analysis for BRLWE is very important since unprotected proposals could be vulnerable to SCA.

Several efforts have been made in the literature to enhance the security of BRLWE against SCA. However, many of these approaches focus only on introducing complex computations to reinforce the algorithm's resilience, often without considering methods to minimize unnecessary computations. In this paper, we present an advanced version of BRLWE that establishes a delicate balance between efficiency and security against timing, SPA and DPA side-channel attacks, while optimizing the overall computation cost. Our approach aims to achieve a more robust and practical approach, addressing the crucial challenge of mitigating side-channel vulnerabilities while maintaining the algorithm's performance. We propose a new random shuffling method designed to enhance the security of the BRLWE algorithm against SCA, specifically against timing SPA and DPA. Moreover, we intend to optimize the polynomial multiplication using the 3-Decomposition Karatsuba multiplication [20], in order to minimize the computation cost. To the best of our knowledge, this combination of the random shuffling method with the 3-Decomposition Karatsuba multiplication, which aims to make a balance between security and efficiency, did not exist in the literature before. We formally prove the security of our approach against quantum hybrid attacks, timing, SPA and DPA attacks. In terms of efficiency, we have evaluated the computation cost, the memory usage and the energy consumption of the proposed approach using a resource-constrained microcontroller, 32-b ARM Cortex-M0. This microcontroller has limited computation resources to prove that our approach can be practical and scalable to even tiny and low-power IoT devices.

The main contributions of this paper are as follows:

1) Enhance the security level of BRLWE using a random shuffling method, which provides timing, SPA and DPA-resistant. Moreover, we provide security analysis against the mentioned attacks and quantum hybrid attacks.

2) Propose a lightweight approach 3DKSh-BRLWE with an optimized polynomial multiplication suitable for resource-constrained IoT nodes, which provides a lower computation time than the original BRLWE.

3) Prove that 3DKSh-BRLWE dominates related works in terms of speed and efficiency on ARM Cortex M0 microcontroller.

The originality of this paper is based on the random shuffling method that is used on the 3-Decomposition Karatsuba multiplication's sub-polynomials to enhance the BRLWE security and efficiency.

The remainder of this paper is organized as follows. In Section 2, we detail the technical background. In Section 3, we present an overview of the existing methods to resist timing, SPA and DPA attacks, as well as to improve the lattice-based security and efficiency. We describe the proposed approach in Section 4. In Section 5, we discuss the security analysis of the proposed approach. We analyze the complexity between the original BRLWE and our proposed 3DKSh-BRLWE approach. In Section 6, we detail the implementation and discuss the results. Finally, we conclude the paper in Section 7.

2. Background

In this section, we present the BRLWE encryption algorithm with uSVP problem that will be used in the security analysis section to evaluate the security level of the proposed approach. Next, we describe the Karatsuba multiplication algorithm. Finally, a notation table summarizing the key symbols and terms used in the paper is provided in Table 1.

2.1 *Binary Ring Learning With Errors (BRLWE)*

The Binary Ring-LWE (BRLWE) is a relatively new variant of the Ring-LWE scheme. It was recently introduced in [9] with rigorous security analysis. The BRLWE scheme uses binary errors instead of Gaussian distributed errors used in the original LWE scheme to reduce the key size and corresponding area complexity. It has great potential for standardization in the future for lightweight applications, despite not being a NIST candidate yet [7]. The BRLWE involves mainly operations over the ring $\mathbb{Z}q/f(x)$, where $f(x) = x^n+1$, and a polynomial of degree $n-1$ with integer coefficients modulo q is considered one typical element in the ring. The scheme can be secured with equivalent class and quantum securities of 190/140 and 84/73 bits, respectively, for $n = 512$ and $n = 256$ [9, 10]. It is based on the hardness of the

Ring-LWE problem, which involves finding the secret key given a set of noisy polynomial equations. The security of the scheme relies on the assumption that the underlying problem is hard to solve, even with the use of quantum computers. The BRLWE-based post-quantum cryptography has three main phases [9], as shown in Figure 2. The scheme in Figure 2 represents a use case of BRLWE in IoT applications, where devices encrypt the collected data and transmit it to the edge or cloud for data analysis or storage after the decryption process.

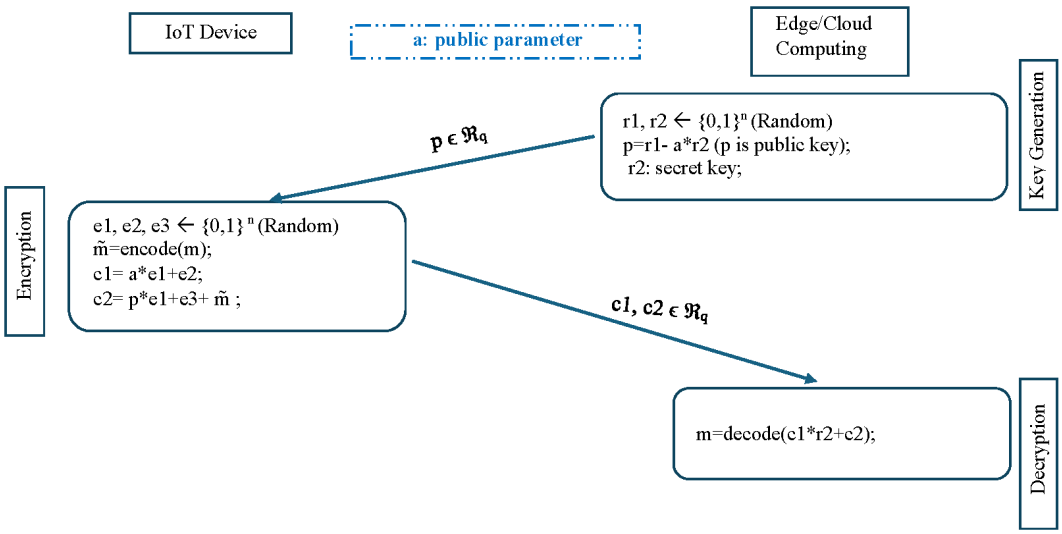

Figure 2. BRLWE phases

Table 1. Notations table

Notation	Description
$\mathbb{Z}q$	Ring of integers modulo q
f(x)	Polynomial defining the ring.
$A \in \mathbb{Z}_q^{mxn}$	Randomly and uniformly selected matrix in the LWE problem.
s	Secret vector.
e	Error vector.
l	Samples in the LWE problem, combining the matrix, secret vector, and error.
$\ell(L)$	Lattice defined by basis L.
α_i	Integer coefficients used in the linear combination of basis vectors in the lattice.
δ	Hermitian delta, used to assess the quality of a lattice basis.
ρ	Success probability of the quantum hybrid attack.
p	Public key.
r_2	Private key.
r_1	Intermediate random polynomial.
Sh( )	Random shuffling function.
c_1, c_2	Ciphertext components.
m	Message to be encrypted.
T_{mul}	Timing complexity of classical polynomial multiplication.
T_{sub}	Timing complexity of polynomial subtraction.
T_{add}	Timing complexity of polynomial addition.
T_{kmul}	Timing complexity of Karatsuba polynomial multiplication.
T_{Sh}	Timing complexity of the shuffling method.

For hardware implementation, as suggested by the authors in [12], coefficients of the polynomials within the ring are adjusted to lie within an inverted range compared to the original. Here, each coefficient is represented in the interval $(-q/2, q/2-1)$, which aligns with the range used in the two's complement representation. This approach allows modular additions and subtractions of coefficients to occur seamlessly, eliminating the need for further reductions. Moreover, we proceed to the LWE problem, the lattice framework, the unique Shortest Vector Problem (uSVP), and the Bounded Distance Decoding problem (BDD). These foundations are essential as we need to convert the LWE problem to the uSVP

problem [21], and then evaluate the security level as described in Section 5. Consider positive integers m, n, and q, where A $\in \mathbb{Z}_q^{mxn}$ represents randomly and uniformly a selected matrix. We denote e $\in \mathbb{Z}_q^m$ as the error vector, drawn from a specific probability distribution, and s $\in \mathbb{Z}_q^n$ as the secret vector. The crux of the LWE problem is to deduce the secret vector s from the given samples (l, A), where l = As + e.

Furthermore, we define the lattice structure and the lattice basis L, which serve as a discrete subset of $\mathbb{R}^m$. A set of vectors constitutes a lattice ℓ if it adheres to the expression:

$$\ell(\mathrm{L}) = \left\{ x \in \mathbb{R}^m \mid x = \sum_{i=1}^{m} \alpha_i l_i, \alpha_i \in \mathbb{Z} \right\}, \tag{1}$$

where x represents any point in the lattice defined by L; m is the dimension of the lattice, indicating the number of linearly independent vectors; and $\alpha_i \in \mathbb{Z}$ are integer coefficients used in the linear combination of the basis vectors l_i. For a linearly independent vector ensemble L = {$l_1, \ldots, l_m$} $\subset \mathbb{R}^m$, such a collection is termed the basis of the lattice ℓ. It is important to note that lattice bases are not unique, encompassing numerous permutations. To assess the quality of Lattice basis L, we use the Hermitian delta δ, with $\|l_1\| = \delta^m \det(\ell)^{1/m}$.

The uSVP challenge involves finding the shortest nonzero vector t in a lattice, which is distinctly shorter than any other vectors that are not integer multiples of t. In contrast, the BDD problem uses a lattice basis L, a distance threshold α, and a target vector t. The goal in BDD is to identify a vector e such that $\|e\| < \alpha\lambda$ and the difference $t - e$ lies within the lattice ℓ.

2.2 *Karatsuba multiplication in BRLWE*

Karatsuba multiplication is a fast algorithm for multiplying two large polynomials, it works by recursively splitting each polynomial into half-size polynomials until the base case of single-digit multiplication is reached. Then, it combines the partial products to obtain the final product. It improves the computational complexity of schoolbook multiplication by reducing the number of elementary operations from $O(n^2)$ to about $O(n^{1.585})$, where n is the number of digits of the operands. It can help optimize memory usage by computing partial products and combining them as needed. This can be advantageous when dealing with limited memory resources or when performing polynomial multiplication on devices with memory constraints [20, 22].

The conventional Schoolbook Polynomial Multiplication Algorithm (SPMA) for two polynomials of length n requires n^2 coefficient products. However, the Karatsuba algorithm can be adapted to polynomial multiplication to reduce the number of coefficient multiplications at the cost of more coefficient additions. This adjustment involves the decomposition of A(x) and B(x) into two distinct components, which are designated by:

$$A(x) = A_1(x) * X^{n/2} + A_0(x) \tag{2}$$

$$B(x) = B_1(x) * X^{n/2} + B_0(x) \tag{3}$$

where A(x) and B(x) are the original polynomials we aim to multiply; X represents the polynomial variable and acts as the placeholder for terms raised to powers of X; n is the length (or degree) of the polynomials; $A_0(x)$ and $B_0(x)$ are the low-order n/2 coefficients of A(x) and B(x); and $A_1(x)$ and $B_1(x)$ are the high-order n/2 coefficients, which are denoted by:

$$A_0(x) = \sum_{i=0}^{n/2-1} a_i X^i, A_1(x) = \sum_{i=0}^{n/2-1} a_{i+n/2} X^i \tag{4}$$

$$B_0(x) = \sum_{i=0}^{n/2-1} b_i X^i, B_1(x) = \sum_{i=0}^{n/2-1} b_{i+n/2} X^i \tag{5}$$

The Karatsuba formula enables computing P(x) and is denoted by:

$$P(x) = x^n * P_2(x) + x^{n/2} * (P_1(x) - P_0(x) - P_2(x)) + P_0(x) \tag{6}$$

where P(x) represents the resulting polynomial product of A(x) and B(x); $P_0(x)$, $P_1(x)$ and $P_2(x)$ are intermediate polynomial products computed as follows:

$$P_0(x) = A_0(x) * B_0(x) \tag{7}$$

$$P_1(x) = (A_0(x) + A_1(x)) * (B_0(x) + B_1(x)) \tag{8}$$

$$P_2(x) = A_1(x) * B_1(x) \tag{9}$$

For conciseness, '(x)' is omitted from the notations if no ambiguity occurs. In (7)–(9), we can see that only three polynomial multiplications of length n/2 are required.

Consequently, these formulas reduce the number of coefficient multiplications from n^2 to $3(n/2)^2 = 3n^2/4$.

3. Related work

In this section, we review the existing lattice-based methods to improve the security and efficiency.

One of the challenges provided by the development of quantum computing is its potential threat to classical public-key encryption schemes. To address this issue, NIST initiated a project to standardize post-quantum cryptography (PQC) algorithms [23]. Out of the many candidates submitted, only four public key encryption and key establishment algorithms have been chosen for the final round: Classic McEliece [24], CRYSTALSKyber [25], NTRU [26], and SABER [27]. These three algorithms are based on hard problems on lattices, such as the closest vector problem (CVP) [28], the shortest vector problem (SVP) [29], and learning with errors (LWE) [10]. Another five algorithms, namely BIKE [30], FrodoKEM [31], HQC, NTRU Prime and SIKE [32], are still under consideration as alternatives.

Lattice-based encryption schemes rely on the hardness of lattice problems, which are believed to be resistant to quantum attacks. Several related works used Lattice cryptography to improve the existing security algorithms to resist quantum attacks. In [33], a new scheme for Ciphertext-Policy Attribute-Based Encryption (CP-ABE) based on the Lattice problem is proposed, in which the authors combined the known CP-ABE algorithm with the Lattice cryptography to resist quantum attacks. The main advantage of this scheme is that it reduces the computation and communication overhead of existing lattice-based CP-ABE schemes, it uses a Linear Secret Sharing Scheme (LSSS) to represent monotone access policies and applies a private key randomization technique to ensure resistance against collision attacks. A variant of pre-quantum RSA called a post-quantum lattice-based RSA (LB-RSA) is proposed in [34] to secure the shared data and information for IoT-based cloud applications. It improves the security level by increasing key dimensions instead of increasing key size. On the other hand, several related works have been proposed to improve Lattice encryption in terms of security and efficiency. In [35, 36] authors improve the complexity and the computation cost of a post-quantum lattice-based SABER algorithm using Number Theoretic Transform (NTT) polynomial multiplication. In [37], a new hardware architecture is proposed to improve the computation cost of post-quantum lattice-based SABER algorithm based on an efficient Karatsuba multiplication algorithm. On the other hand, Lyubashevsky et al. [10] introduced a new variant of the Learning With Errors (LWE) problem that applies the ideal lattices in the LWE, this variant is called ring-LWE. The

polynomial ring is used for all operations and the error polynomials are sampled from a discrete Gaussian distribution. Ring-LWE reduces the public key and the private key size by n times while keeping the same security level as standard LWE instances. The ring-LWE has been used for key exchange, homomorphic encryption, digital signature, public key encryption, etc. However, it faces challenges in efficient deployment due to the complexity of its Gaussian sampler and the high cost of polynomial multiplication in terms of time and resources [38]. A shuffling method is proposed in [39] to avoid the expensive constant-time Gaussian sampler. Many hardware and software implementations in the literature aim to optimize these aspects of polynomial multiplication. Howe et al. [40] presented a comprehensive evaluation of different methods for sampling from a discrete Gaussian distribution. They proposed novel optimized hardware architectures for several sampling techniques, such as the Cumulative Distribution Table (CDT), the Bernoulli, the Knuth-Yao and the discrete Ziggurat samplers. Moreover, the authors presented the first constant-time hardware designs for some of these samplers, which offer protection against timing analysis. They concluded that the CDT sampler is the most suitable for a balanced performance, achieving a high throughput and a low area consumption for both encryption and signature applications. Another work that tries to improve the Ring LWE is the Binary Ring LWE (BRLWE) [9]. The authors proposed a new public-key encryption scheme using the binary noise distribution instead of the Gaussian one, which reduces the size of the ciphertexts and the computational cost of encryption and decryption. It presents efficient implementations of the scheme on 8-bit and 32-bit microcontrollers, which are suitable for low-power and low-cost scenarios such as the IoT.

Based on BRLWE, Aydin et al. [14] proposed a hardware implementation of BRLWE along with low-cost power side-channel countermeasures, they use a randomization of intermediate states and masked threshold decoding to improve the security against DPA. Another hardware implementation of BRLWE is presented in [41], it is based on masking countermeasures against differential power analysis (DPA) attacks without employing any pseudo-random number generator (PRNG) module on lightweight implementations, which improves the speed and efficiency. In [42], authors proposed an efficient hardware architecture for the BRLWE-based cryptographic scheme, targeting lightweight applications. The architecture computes the arithmetic operation AB + C, which includes polynomial multiplication and addition over the polynomial ring. It reduces resource utilization, increases frequency, and lowers power consumption compared to existing BRLWE-based schemes. Ebrahimi et al. [17] proposed fault-resilient implementations of BRLWE for IoT Devices. The paper analyzes the vulnerability of previous BRLWE implementations to different types of fault attacks, such as randomization, zeroing, and skipping faults, and shows how they can be easily broken by an adversary. Moreover, it proposes new fault-resilient software implementations of BRLWE on two resource-constrained microcontrollers, namely AVR ATxmega128A1 and ARM Cortex-M0, which are suitable for IoT devices.

A variant of Ring-LWE, known as the Ring-ExpLWE scheme [21], has been introduced, where the error vector is drawn from an exponential distribution. This makes the noise distribution more discrete and increases the security level of the scheme. Its security level is evaluated by comparing and analyzing the runtime of BRLWE and Ring-ExpLWE when subjected to quantum hybrid attacks, as well as examining the noise polynomials in each scheme. The Ring-ExpLWE scheme has a high performance and a low area-time product compared to BRLWE while maintaining a strong security level.

All the previous proposed works that aim to improve the BRLWE try to improve the security of the algorithm against SCA by adding more computation tasks. They tried to present lightweight methods, but it was still not enough. The computation time was always higher than the computation time of the original BRLWE. In this paper, we aim to enhance the security of the BRLWE algorithm against SCA, including timing, SPA, and DPA, and to optimize the polynomial multiplication process which is a complex task within the BRLWE algorithm. The combination of adding computation tasks to improve the security and at the same time optimize the polynomial multiplication helps us to create a balance between security efficiency.

4. Proposed approach

In this section, we present the different variants of Karatsuba multiplication algorithms then, we display the new shuffling method-based Karatsuba multiplication and how it enhances the security of BRLWE and improves its computation cost. Moreover, we exhibit the main approach including a detailed description of each phase of the proposed BRLWE.

4.1 *Karatsuba polynomial multiplication*

Figure 2 depicts a summary of all the arithmetic operations involved with each phase of the BRLWE. In the key generation phase, polynomial multiplication and polynomial subtraction are performed to generate the public key p. The encryption phase involves polynomial multiplication and polynomial addition to compute the first part of the ciphertext, c1, and requires additional polynomial multiplication and two polynomial additions to generate c2. Finally, in the decryption phase, polynomial multiplication and polynomial addition are used to recover the original message.

We can see that the major arithmetic complexity of each phase of the BRLWE scheme lies mainly in the polynomial multiplication (Polynomial addition and substruction are just point-wise operations). To improve the BRLWE computation cost we need to optimize the polynomial multiplication.

SPMA is a highly efficient method for multiplying polynomials due to its straightforward computational structure. Its folded architecture is simple to implement, consumes a small hardware area and has a low throughput performance. The complexity of the SPMA is $O(n^2)$. On the other hand, the NTT is an advanced method that can achieve high-speed polynomial multiplication with a complexity of O(nlogn). However, NTT requires complicated pre-computation and array re-ordering to achieve high speed, requiring much higher memory usage and area consumption compared to SPMA.

The Karatsuba algorithm offers a middle ground between these two extremes. It has a computational complexity of $O(n^{\log_2 3})$, which is lower than SPMA but higher than NTT. This makes Karatsuba a favorable choice for balanced implementations of resources-constrained Lattice-Based Cryptography (LBC). Furthermore, in the round 3 submission to the NIST Post-Quantum Cryptography (PQC) standardization process, the authors of the Saber cryptographic scheme advised against using NTT for polynomial convolution. The primary reason is that the ring structure used in Saber, typically does not facilitate a straightforward implementation of NTT due to the difficulty in finding appropriate roots of unity and ensuring compatibility with the modulus [37]. Similarly, in the context of BRLWE, which often involves rings with binary coefficients, implementing NTT can be challenging. The absence of suitable roots of unity and the specific modulus requirements make NTT impractical for such schemes. Therefore, alternatives like Karatsuba are preferred for their more manageable complexity and compatibility with the ring structures used in BRLWE.

Besides the reduced computation cost of Karatsuba and its simplicity to implement, it can provide another layer of security for BRLWE. By recursively splitting polynomials and shuffling each sub-polynomial separately, even if an attacker discovers the shuffling factors, they will only gain partial information about the secret key rather than the entire key.

4.1.1 *Decompositions of karatsuba multiplication*

In Karatsuba multiplication, increasing the decomposition factor can further decrease the number of coefficient multiplications, thus lowering computational costs. When A(x) and B(x) are divided into three segments, each of length n/3 rather than two, their product P(x) can be represented as:

$$P(x) = P_0(x) + P_1(x)x^{n/3} + P_2(x)x^{2n/3} + P_3(x)x^n + P_4(x)x^{4n/3} \tag{10}$$

It can be computed according to [43] as follows:

$$P_0(x) = A_0(x)B_0(x) \tag{11}$$

$$P_1(x) = (A_0(x) + A_1(x))(B_0(x) + B_1(x)) - A_1(x)B_1(x) - A_0(x)B_0(x) \tag{12}$$

$$P_2(x) = (A_0(x) + A_1(x) + A_2(x))(B_0(x) + B_1(x) + B_1(x)) - ((A_0(x) + A_1(x))(B_0(x) + B_1(x)) - A_1(x)B_1(x)) - ((A_1(x) + A_2(x))(B_1(x) + B_2(x)) - A_1(x)B_1(x)) \tag{13}$$

$$P_3(x) = (A_1(x) + A_2(x))(B_1(x) + B_2(x)) - A_1(x)B_1(x) - A_2(x)B_2(x) \tag{14}$$

$$P_4(x) = A_2(x)B_2(x) \tag{15}$$

Using three segments, six multiplications between polynomials of length n/3 are required, leading to a total of $6(n/3)^2 = 2n^2/3$ coefficient multiplications. For a four-segment decomposition, ten multiplications are needed between polynomials of length n/4, resulting in $10(n/4)^2 = 5n^2/8$ coefficient multiplications. The formulas for two-segment decompositions, as shown in (6)–(9), and for three-segment decompositions in (10) through (15), can be iteratively applied to achieve higher decompositions and further minimize the number of coefficient multiplications.

4.1.2 *Karatsuba polynomial multiplication with integrated modular reduction*

The Karatsuba algorithm aims to reduce the number of coefficient multiplications, but the intermediate segment products must be assembled appropriately to compose the final polynomial product. This assembly process needs more storage for intermediate results and supplementary additions and subtractions. These can cause problems, especially for applications with limited memory.

The works [44, 43] proposed to apply the integrated modular reduction by x^n+1 directly on the segment products during Karatsuba multiplication before adding them up. This technique optimizes the assembly process before the final addition/ substruction of segments, thereby reducing the number of additions/ substructions without increasing multiplication complexity. Furthermore, the shared terms help to minimize the memory size, which stores intermediate results and accounts for a significant portion of the silicon area required for polynomial multiplication as shown in Figure 3.

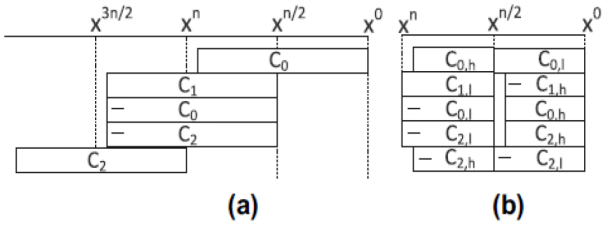

Figure 3. Segment products needed to be added up in Karatsuba multiplication with 2-decomposition: (**a**) original; (**b**) with integrated modular reduction by $x^n + 1$ [44]

4.2 *The proposed Karatsuba multiplication based random shuffling method*

The shuffling method aims to add randomness through permitting operations order. It effectively hides the correlation between the secret key and the ciphertext, making it harder for an attacker to extract information. The shuffling method provides strong security guarantees and can withstand attacks based on statistical analysis [45, 46]. However, typically,

it produces additional computation costs and memory usage. The effectiveness of the shuffling method and the exact computation cost relies on the shuffling algorithm used.

In this paper, we propose to use the Fisher-Yates Shuffle algorithm known as the Knuth shuffle [45]. It is a widely used algorithm for shuffling a finite sequence, due to its simplicity to understand and implement. This algorithm shuffles the element of the array in place without requiring additional memory or data structures and performs a linear number of permutations that make it efficient for IoT applications. Moreover, it produces uniformly random permutations and provides good randomness proprieties which make it efficient against Timing and SPA attacks.

However, in [45], the authors proved that the simple shuffling algorithm can not protect against DPA attacks. These attacks exploit the correlations in power consumption between different clock cycles, making it possible to infer sensitive information despite the simple shuffling.

To eliminate these correlations between the power consumption and the sensitive information, we propose to divide the polynomials into sub-polynomials during the polynomial multiplication and randomly shuffle the order of operations of each sub-polynomial multiplication. Each multiplication involves several smaller operations, and because these operations are performed in a random order, the overall power consumption pattern becomes more complex and less correlated with any specific sub-operation. Which reduces the ability of an attacker to combine information from different parts of the trace to deduce the key.

We propose to apply Karatsuba multiplication algorithm in polynomial multiplication. This algorithm is based on recursively splitting each polynomial into n-size sub-polynomials then combining the partial products to obtain the final product. The main objective of this algorithm is to optimize the polynomial multiplication cost, which improves the computation cost of the random shuffling method.

To determine the most efficient Karatsuba multiplication algorithm to enhance the computation cost and security of BRLWE. We implement the simple Karatsuba multiplication (2-Decompositions) based shuffling for BRLWE, 3-Decompositions Karatsuba based shuffling for BRLWE, 4-Decompositions Karatsuba based shuffling for BRLWE and 3-Decompositions Karatsuba with integrated modular reduction-based shuffling for BRLWE, as shown in Table 3 in Section 5. According to the results presented in Table 3, the 3-decomposition Karatsuba-based shuffling for BRLWE gives the best results. Algorithm 1 represents the proposed 3-Decomposition Karatsuba-based random shuffling method.

Generally, Karatsuba multiplication is particularly effective for multiplying large numbers. It gives better results than the SPMA only with big numbers because it requires more additions and digit shifts. For this reason, as illustrated in Algorithm 1, we apply the Karatsuba algorithm only when the polynomial degree is more than 32.

Furthermore, to introduce additional noise and enhance randomization and to cancel the correlation between clock cycles, we propose to refresh the shuffling factor in each cycle. During each cycle, a new random number is used for shuffling, breaking the link between consecutive clock cycles. As a result, potential DPA attackers cannot extract any meaningful information from trace analysis across different cycles. By applying this approach, the shuffling method effectively obscures the actual sequence of operations, making it significantly more challenging for attackers to deduce sensitive information from side-channel leakage.

The polynomial splitting and the updates of the shuffling factor in each cycle provide an added layer of security, ensuring the resilience of the cryptographic scheme against DPA attacks.

Algorithm 2 exhibits the random shuffling algorithm for each sub-polynomial. The proposed random shuffling algorithm generates a random number r between <1, n/3−k>, called shuffling factor, where n is the length of the polynomial in each cycle and k is the number of elements already processed in the cycle. We divide n by 3 because we used 3-Decomposition Karatsuba multiplication (see (14), (15), (17) and (18)), and we shuffle each sub-polynomial separately to add more noise and make it harder for an attacker to predict the whole polynomial of the secret key. In the first iteration, the algorithm swaps the last element of the sub-polynomial with the r-th element, and the iteration continues to reduce the size of the sub-polynomial by 1 until the first element is reached. This made the time complexity O(n) with space complexity O(1) for the random function.

Algorithm 1 3-Decompositions Karatsuba based Shuffling

Input: $A(x), B(x) \leftarrow \{0,1\}^n$, n is power of 3
Output: $P(x) = A(x) \cdot B(x)$
$N \leftarrow \max(\text{degree}(A), \text{degree}(B))$
if $n = 1$ **then**
    $P(x) \leftarrow A(x) \cdot B(x)$
else if $n < 32$ **then**
    $P(x) \leftarrow$ Schoolbook Polynomial Multiplication(A, B)
else
    Pre-processing Phase
    $m \leftarrow n/3$
    $A_0(x) \leftarrow A_0(x)[m-1:0]$;
    $A_1(x) \leftarrow A_1(x)[2m-1:m]$;
    $A_2(x) \leftarrow A_2(x)[n-1:2m]$;
    $B_0(x) \leftarrow B_0(x)[m-1:0]$;
    $B_1(x) \leftarrow B_1(x)[2m-1:m]$;
    $B_2(x) \leftarrow B_2(x)[n-1:2m]$;
    Shuffling Phase
    $A'_0(x) \leftarrow$ Random Shuffling$(A_0(x), m)$;
    $A'_1(x) \leftarrow$ Random Shuffling$(A_1(x), m)$;
    $A'_2(x) \leftarrow$ Random Shuffling$(A_2(x), m)$;
    $B'_0(x) \leftarrow$ Random Shuffling$(B_0(x), m)$;
    $B'_1(x) \leftarrow$ Random Shuffling$(B_1(x), m)$;
    $B'_2(x) \leftarrow$ Random Shuffling$(B_2(x), m)$;
    Recursive and Polynomial multiplication
    $P_0(x) \leftarrow$ Karatsuba$(A'_0(x), B'_0(x), m)$;
    $P_1(x) \leftarrow$ Karatsuba$(A'_0(x) + A'_1(x), B'_0(x) + B'_1(x), m)$;
    $P_2(x) \leftarrow$ Karatsuba$(A'_0(x) + A'_1(x) + A'_2(x), B'_0(x) + B'_1(x) + B'_2(x), m)$;
    $P_3(x) \leftarrow$ Karatsuba$(A'_1(x) + A'_2(x), B'_1(x) + B'_2(x), m)$;
    $P_4(x) \leftarrow$ Karatsuba$(A'_2(x), B'_2(x), m)$;
    $P_5(x) \leftarrow$ Karatsuba$(A'_2(x), B'_1(x), m)$;
    Post-processing Phase
    $P_1(x) \leftarrow P_1(x) - P_5(x) - P_0(x)$;
    $P_2(x) \leftarrow P_2(x) - (P_1(x) - P_5(x)) - (P_3(x) - P_5(x))$;
    $P_3(x) \leftarrow P_3(x) - P_5(x) - P_4(x)$;
    $P(x) \leftarrow x^{4n/3}P_4(x) + x^n P_3(x) + x^{2n/3}P_2(x) + x^{n/3}P_1(x) + P_0(x)$;
end if
return $P(x)$

Algorithm 2 The Proposed Random Shuffling Algorithm

Input: sub-polynomial of coefficients c, sub-polynomial degree m
Output: Shuffling c
for each Karatsuba cycle **do**
  $m \leftarrow n/3$
  for j from $(m-1)$ down to 1 **do**
    random integer r between 0 and j
    Swap $c[j]$ and $c[r]$
  end for
end for
return c

4.3 *Main approach*

In this paper, we focus primarily on data confidentiality and encryption to secure transmitted data within the 3DKSh-BRLWE scheme. We assume that IoT devices are pre-authenticated on the Cloud/Edge side, and the Cloud/ Edge infrastructure itself is treated as a secure entity. This assumption allows us to ensure that data is only encrypted and transmitted by verified devices, while enabling our efforts to be dedicated to developing a robust encryption approach to protect data confidentiality during transmission.

In this section, we present the main components and functions of the proposed robust encryption approach, as shown in Figure 4. First, we present the two components of the proposed approach which are: IoT device and cloud/edge computing.

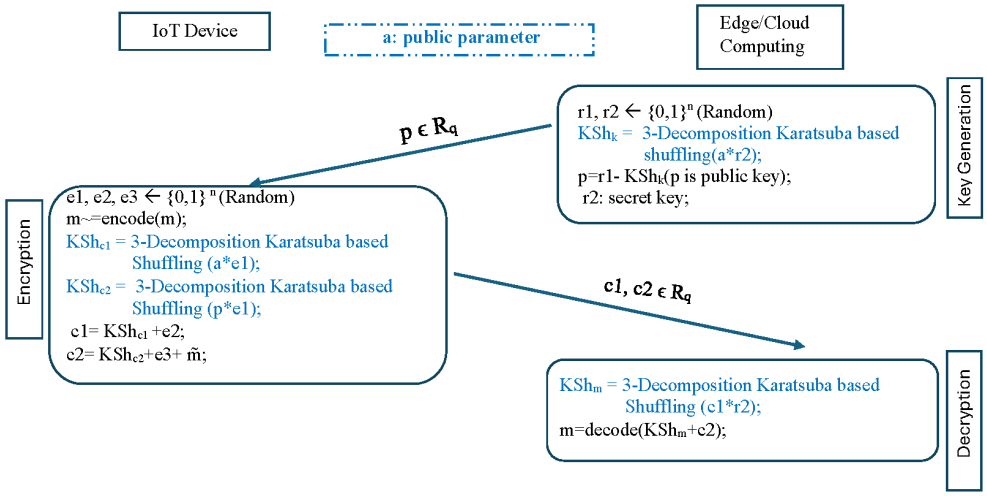

Figure 4. The proposed 3DKSh-BRLWE phases

(1) Cloud/Edge COMPUTING

This entity is responsible for generating both the public and private keys. It generates and securely stores the private key, while the public key is shared with authenticated devices. Additionally, cloud/edge computing performs decryption operations using the private key, which is retained solely within the cloud infrastructure and not distributed.

(2) IoT Device

This entity is tasked with encrypting the data it collects using the public key provided by the cloud.

Second, we present the main three functions of the proposed approach which are, key generation, encryption and decryption:

(1) Key generation

This phase is responsible for generating the public and the private keys. In this phase, two binary polynomials, r1 and r2, are randomly selected, where r2 is the secret key. Therefore, the main operation is to produce the public key using (10).

$$p = r1 - Sh(a * r2), \tag{16}$$

where a is the public polynomial of the integer, Sh( ) is the random shuffling function, and the operation (*) is the 3-Decomposition Karatsuba polynomial multiplication. The size of public key p and private key r2 is different, private key r2 is composed of n bits while the public key p consists of n_log2 q bits, and the polynomial r1 will be ignored after the public key calculation. To optimize the polynomial multiplication and to protect the secret key from timing and power analysis attacks (SPA and DPA), we apply a 3-Decomposition Karatsuba-based random shuffling on the public key computation to protect sensitive information such as a secret key.

(2) Encryption

This phase is responsible for the encryption of n-bit message m. Firstly, the message is mapped to a polynomial m $\in \mathscr{R}$q using (17), and secondly, it generates the ciphertexts c1 and c2 from the message using (18) and (19). The polynomials c1 and c2 are computed using the public key (p) and the random vector errors e1, e2 and e3 $\in \mathscr{R}_q$ are uniformly and randomly selected from $\{0,1\}^n$. To add noise and reduce the computation time of the ciphertexts, we apply the proposed 3DKSh algorithm on c1 and c2 computation. We apply the 3-Decomposition Karatsuba multiplication (*) and randomly shuffle (Sh) each sub-polynomials in the polynomial multiplication of c1 and c2.

$$m = ENCODE\ (m)$$

$$(m_0, \ldots, m_{n-I}) \rightarrow \sum_{i=0}^{n-1} m_i \left(-\frac{q}{2}\right) x^i \tag{17}$$

where (m_0,…,m_{n-1}) represents the message as a vector of coefficients, m_i denotes the i-th coefficient of the message, q is a prime number related to the modulus used in the cryptographic scheme and x^i represents the power of x corresponding to the position of each coefficient.

$$c1 = Sh(a * e1) + e2 \tag{18}$$

$$c2 = Sh(p * e1) + e3 + m \tag{19}$$

It is worth noting that in (17), we use the new variant proposed by [15]. This variant used the two's complement range to manage the reduction operation and eliminate individual reduction checks using natural overflow. The range of polynomial coefficient values is adjusted from $(-\lfloor q/2 \rfloor + 1, \lfloor q/2 \rfloor)$ to $(-\lfloor q/2 \rfloor, \lfloor q/2 \rfloor - 1)$.

(3) Decryption

This phase is responsible for ciphertext decryption and message recovery. It uses the secret key r2 and the ciphertexts c1 and c2 to generate the decryption polynomial d $\in \mathscr{R}_q$ calculated by (20). In this phase, we also apply the proposed 3DKSh

algorithm in the polynomial multiplication of c1 and r2. Using the random shuffling method (Sh) and the 3-Decomposition Karatsuba multiplication (*), to protect the secret key while reducing the computation time. The threshold decoder in (22) processes each coefficient separately. It outputs a binary '0' if the calculated polynomial's coefficient falls outside the range of (q/4, 3q/4), otherwise, it outputs a '1'. The application of the BRLWE decryption function directly can cause half of the decoding failure, thus it is necessary to take into consideration the boundary overflow problem under the discrimination condition [15].

$$d = Sh(c1 * r2) + c2 \tag{20}$$

DECODE : $\mathscr{R}_{\mathrm{q}} \rightarrow \{0, 1\}^{\mathrm{n}}$

$$\sum_{i=0}^{n-1} d_i x^i \rightarrow (m_0 \ldots m_{n-1}), \tag{21}$$

$$m_i = \begin{cases} 0, |d_i - i - \lfloor \frac{n-3}{2} \rfloor| > \frac{q}{4} \\ 1, \text{else} \end{cases} \tag{22}$$

5. Security analysis

In this section, we formally analyze the security of 3DKSh-BRLE against quantum hybrid attacks and timing, SPA and DPA attacks.

5.1 *Quantum hybrid attack*

To analyze the security level of our approach against a quantum hybrid attack and compare it with BRLWE, we evaluate the runtime for both schemes under the same parameter set. Initially, we explore the runtime for our approach using a quantum hybrid attack, which is suitable for evaluating both BRLWE and our enhanced version due to their sparse secret and error vectors [10]. Hybrid attacks consist of classical and quantum versions [47]. The quantum hybrid attack replaces the classical meet-in-the-middle technique [48] with a quantum search algorithm for faster guessing. In the quantum hybrid attack [47], we first convert the LWE instance into a uSVP problem using (23):

$$\ell = \left\{ x \, \varepsilon \, \mathbb{Z}^d \; : \; (A \; |I_m| \; - \; b) \, x = \; 0 \bmod q \right\} \tag{23}$$

where ℓ is a d-dimensional lattice, in which d = n + m + 1; $(A \; |I_m| \; - \; b)$ is an augmented matrix that includes: A$\varepsilon \; \mathbb{Z}_q^{m \times n}$ a randomly chosen matrix representing the public part of the LWE instance, I_m the identity matrix of size m × m, which ensures that the error vector e is incorporated into the lattice structure; and $-$b which is a column vector representing the negation of the known vector b from the LWE instance. This component links the lattice definition directly to the original LWE instance. The vector v = (s, e, 1) is an element of the lattice ℓ, because of s $\in \; \mathbb{Z}_q^n$ and e $\in \; \mathbb{Z}_q^m$.

The second step is to solve the uSVP problem using (17):

$$\acute{\mathrm{L}} = \begin{pmatrix} L & C \\ 0 & I_r \end{pmatrix} \in \mathbb{Z}^{d \times d}, \tag{24}$$

where $\acute{\mathrm{L}}$ is a basis for lattice ℓ, L $\in \; \mathbb{Z}^{(\mathrm{d\text{-}r}) \times (\mathrm{d\text{-}r})}$, r $\in \; \mathbb{N}$ and r < m, n. We can represent the short vector v = ($\mathrm{v_l}$, $\mathrm{v_g}$) by lattice basis $\acute{\mathrm{L}}$ as it belongs to the lattice ℓ using (15):

$$v = \begin{pmatrix} v_l \\ v_g \end{pmatrix} = \acute{\mathrm{L}} \begin{pmatrix} h \\ v_g \end{pmatrix} = \begin{pmatrix} Lh + CV_g \\ v_g \end{pmatrix} \tag{25}$$

where $\mathrm{Lh} \in \mathbb{Z}^{\mathrm{d-r}}$ is a vector in Lattice $\ell' \in \mathbb{Z}^{\mathrm{d-r}}$ and $\mathrm{v_l} - \mathrm{Lh} = \mathrm{Cv_g}$, $\mathrm{h} \in \mathbb{Z}^{\mathrm{d-r}}$. Therefore, by solving the BDD problem with $\mathrm{Cv_g}$ as the target vector, we can recover the vector $\mathrm{v_l} \in \mathbb{Z}^{\mathrm{d-r}}$, provided that $\mathrm{v_g} \in \mathbb{Z}^{\mathrm{r}}$ is guessed. To enhance the efficiency of solving the BDD problem using the Nearest Plane algorithm [49], it is necessary to precompute a high-quality lattice basis $\mathrm{L} \in \mathbb{Z}^{(\mathrm{d-r}) \times (\mathrm{d-r})}$ for the Lattice ℓ. The total runtime of the quantum hybrid attack is denoted by:

$$T = \frac{T_h + T_r}{\rho}, \tag{26}$$

where $\mathrm{Th} = \mathrm{l} \cdot (\mathrm{d} - \mathrm{r})^2 / 2^{1.06}$ is the runtime for the hybrid attack, Tr is the lattice reduction runtime and ρ is the quantum hybrid attack success probability [47], which is designated by:

$$\rho \approx \prod_{i=0}^{d-r} \left(1 - \frac{2}{B\left(\frac{(b-r)-1}{2}, \frac{1}{2}\right)} \int_{-1}^{\max(-r_i - 1)} \left(1 - t^2\right)^{\frac{(d-r-r)-3}{2}} dt \right), \tag{27}$$

where $\mathrm{B}(\bullet, \bullet)$ denotes the Euler beta function. We define $\mathrm{r_i}$ as $\mathrm{R_i}/2 \, \|v_l\|$, where $\mathrm{i} \in \{1, \ldots, \mathrm{d-r}\}$ and $\mathrm{R_i}$ is the length of the Gram-Schmidt basis vectors [47] corresponding to the basis L. Here, $\|v_l\|$ is the Euclidean norm of the vector $\mathrm{v_l}$. The runtime of lattice basis reduction depends on both the basis quality δ the lattice dimension $\mathrm{d-r}$. Therefore, optimizing the total runtime for a quantum hybrid attack is dependent on the attack parameters r and δ. Hence, it is crucial to optimize the total runtime across all possible choices of attack parameters is crucial for performance. Given the same attack parameters (r, δ) in both schemes, the runtime $\mathrm{T_r}$ for BRLWE and our proposal remains consistent since their lattice dimensions remain the same. $\mathrm{T_h}$ denotes the runtime of the hybrid attack, where l is the number of guessing cycles for $\mathrm{v_g}$. The computational complexity of a single run of the Nearest Plane algorithm is $(\mathrm{d} - \mathrm{r})^2 / 2^{1.06}$. Based on this analysis, the Nearest Plane algorithm's runtime is equivalent in both our scheme and BRLWE, given the same d and r. For a structured search space, Q-search yields more efficient search cycles. When distributed over a set $\mathrm{S} = \{-16, \ldots, 16\}^{\mathrm{r}}$, using the more generalized Grover's search algorithm [48], the search cycles l is designated by:

$$l = \left(\left(\sum_{i=0}^{32} \rho^{\frac{3}{2}} i \right)^r \right)^{\frac{3}{2}} \approx 2^{1.85r}, \; \rho_i = \binom{32}{i} 2^{-32} \tag{28}$$

We apply a generalized version of Grover's search algorithm across both schemes, resulting in identical cycles required for guessing the vector $\mathrm{v_g}$. Thus, the runtime $\mathrm{T_h}$ remains the same for both schemes. However, when evaluating the success probability ρ, it's crucial to recognize that the vector $\mathrm{v_l}$ in our approach exhibits a larger Euclidean norm compared to BRLWE. This is due to the error vector e. In our approach, we hide the error vector by shuffling the order of operations and refreshing the shuffling factor which adds more noise. As a result, when computing the integral in the attack success probability (27), the value of $\mathrm{r_i}$ for our scheme is smaller than that of BRLWE. As a result, the integral term in the success probability ρ increases for our approach due the upper bound of the integral being-$\mathrm{r_i}$. This increase in the integral's result causes each factor in the continuous multiplication in Equation (27) to be smaller, thereby reducing the quantum hybrid attack's success probability for our scheme relative to BRLWE, and consequently leading to a larger total runtime. Furthermore, this conclusion assumes that both schemes employ identical attack parameters. In a more general scenario, when both schemes are evaluated across the complete set of attack parameters, the total runtime of our approach is higher than that of BRLWE (before applying 3-Decopsition Karatsuba algorithm). Thus, our approach exhibits higher runtime complexity, before reducing the polynomial multiplication cost, when optimized on the attack parameter set. This observation also indicates that it offers a stronger security level compared to BRLWE.

5.2 *Timing and power analysis countermeasures*

An encryption scheme is considered robust and reliable when it not only provides sufficient security at the algorithmic level but also demonstrates resistance to side-channel attacks [48]. NIST specifically requires practical evaluations of submitted schemes for resilience against such attacks, especially in IoT infrastructures where long-term security is critical [23]. Attackers often have physical access to devices that store secret cryptographic keys. Without specific countermeasures, side-channel attacks, such as timing and power analysis, can be exploited to extract these keys, compromising the security of the entire system.

In this section, we discuss how timing, SPA and DPA side channel attacks can infect the BRLWE and how can our enhanced approach improve the security to resist these attacks. Precisely, we analyze the decryption phase because of the direct dependency of each round's behavior on the secret key. In the proposed scheme, the parameter set (n, q) offers flexibility and extensibility. By increasing the values of n and q, the security level improves, when (n, q) takes the value (256, 256), the scheme achieves 84-bit and 73-bit classical and quantum security, respectively. On the other hand, for (n, q) = (512, 256), the security level reaches 190-bit and 140-bit, respectively [9, 10]. However, increasing the values of n and q demands more computing resources.

In our approach, and because we use the 3-Decomposition Karatsuba polynomial multiplication, the polynomials of the ciphertext and the secret key are divided into three sub-polynomials with degree n/3. The polynomial multiplication is computed using (11)–(15). To explain how our approach can resist timing, SPA and DPA attacks, we give the following example:

C1= (90,50,30,120,80,150,20,60,70,59,85,94)

R2 = (1,1,1,0,1,0,1,1,0,0,1,1)

Two polynomials with n = 12 and q = 256. First, we split the polynomials into three sub-polynomials with n = 4 and q = 85, which gives: $C1_0$ = (90,50,30,120), $C1_1$ = (80,150,20,60), $C1_2$ = (70,59,85,94), $R2_0$ = (1,1,0,1), $R2_1$ = (1,1,1,0) and $R2_2$ = (0,0,1,1).

Second, we randomly shuffle each sub-polynomial using different shuffling factors to eliminate the correlation between the sub-polynomials, where: $C1_0$ = (50,90,120,30), $C1_1$ = (60,150,80,20), $C1_2$ = (85,94,70,59), $R2_0$ = (1,1,0,1), $R2_1$ = (1,1,0,1) and $R2_2$ = (0,1,0,1).

By using these steps, we added a new security layer against Timing, SPA and DPA attacks by eliminating any correlation between the sub-polynomials. Consequently, if an attacker manages to determine the shuffling factor of a sub-polynomial multiplication, they would only be able to detect a portion of the secret and not the entire secret message. This significantly enhances the security of the BRLWE scheme by compartmentalizing the information and making it substantially more difficult for an attacker to reconstruct the full secret.

To eliminate the correlation during the polynomial multiplication, we randomly shuffle each multiplication cycle using different shuffling factors. Figure 5 shows the polynomial multiplication of the first sub-polynomials $C1_0$ and $R2_0$.

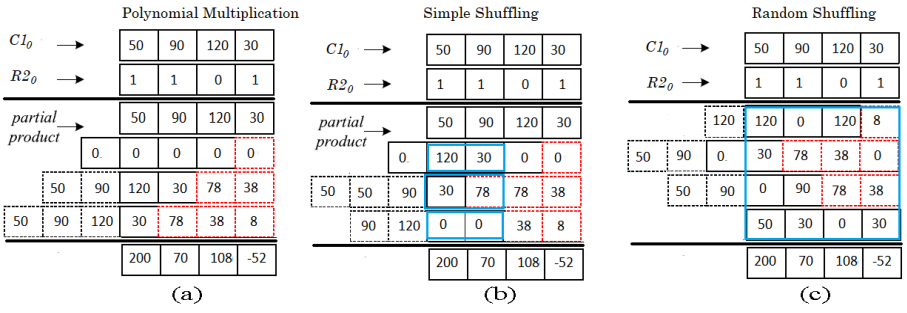

Figure 5. The difference between: (**a**) simple polynomial multiplication. (**b**) polynomial multiplication with simple shuffling. (**c**) polynomial multiplication with random shuffling

The calculation involving the private key $R2_0$ entails polynomial multiplication. Notably, when $R2_0$ is 0, the corresponding partial product becomes 0, as shown in Figure 5a. A crucial concern arises when performing the addition operation directly, as one input of the adder is 0. This leads to distinct spikes in the measured power trace, which allows the adversary to analyze the power differences caused by these operations. Yet, the attacker could attempt a Simple Power Analysis (SPA) attack, extracting the private key with only a few measurements. Such attacks rely on interpreting power trace variations due to specific operations like addition and memory updates, which do not occur by default.

Using the shuffling method enables the manipulation of the order of operations during polynomial multiplication without affecting the result. This introduces randomness and confusion, making it significantly more challenging for adversaries to extract sensitive information from side-channel leakage. Moreover, using Karatsuba multiplication and especially dividing the polynomial into three sub-polynomial and shuffling the sub-polynomial multiplications will add more noise and protect the secret key.

As shown in Figure 5a, each addition row is computed in one cycle, so to generate the first-row addition it will compute: (30 + 0), (120 + 0), (90 + 0), and (50 + 0). Using the simple shuffling method, as shown in Figure 5b, the first-row addition can be generated as follows: (30 + 0), (120 + 0), (90 + 30), and (50 + 120). This adds confusion to the attacker to detect the position of the zero and in which the vector exists. Here, our proposed approach resists the simple power analysis (SPA) attacks. Furthermore, it enhances resistance against timing attacks. Each clock cycle now yields different operations, disrupting any correlation between the computation time of each cycle and the processed data. In the example illustrated in Figure 5a, the computation time for each cycle might be distinguishable due to the consistent order of operations in each cycle. The shuffling method, however, disrupts this regularity, making the timing attack ineffective in capturing meaningful timing differences.

On the other hand, the simple shuffling method has been shown vulnerable to DPA attack, as demonstrated in [45]. This differential attack leverages a random input during decryption and formulates hypotheses on an intermediate addition. Subsequently, the attack validates these hypotheses through power trace analysis on each cycle, enabling the detection of the shuffling factor's value and facilitating the recovery of the first part of the secret key ($R2_0$).

Therefore, it is hard for a DPA attacker to make a hypothesis and recover the two other parts of the secret key ($R2_1$), ($R2_2$) but it is still possible. In response to this DPA vulnerability, we propose to refresh the shuffling factor's value in each cycle in sub-polynomial multiplication. This strategic refreshment aims to break any correlation between consecutive cycles, rendering hypothesis-making and power trace analysis ineffective. The constant alteration of the shuffling factor during each cycle completely changes the order of operations, preventing any meaningful correlation between consecutive cycles, as shown in Figure 5c. Consequently, the lack of correlation between the shuffling factors in successive cycles introduces substantial noise, effectively thwarting the DPA attacker's efforts to recover the first sub-polynomial of the secret key $R2_0$.

5.3 *Complexity analysis*

In this section, we present a comprehensive comparison between the original BRLE scheme and our proposed enhanced version, 3DKSh-BRLWE in terms of complexity analysis. As illustrated in Table 2, our 3DKSh-BRLWE presents better complexity than the original BRLWE even if we added the shuffling computation. This is because we have optimized the most costing operation (i.e., polynomial multiplication).

Table 2. Time complexity analysis of the proposed approach

Phases	BRLWE	3DKSh-BRLWE
Key generation	$T_{mul} + T_{sub} \rightarrow O(n^2)$	$Tk_{mul} + xT_{Sh} + T_{sub} \rightarrow O(n^{\log_3(6)})$
Encryption	$2T_{mul} + 3T_{add} \rightarrow 2O(n^2)$	$2\,Tk_{mul} + xT_{Sh} + 3T_{add} \rightarrow 2O(n^{\log_3(6)})$
Decryption	$T_{mul} + T_{add} \rightarrow O(n^2)$	$Tk_{mul} + xT_{Sh} + T_{add} \rightarrow O(n^{\log_3(6)})$

T_{mul}, T_{sub}, T_{add}, T_{kmul}, T_{Sh}, represent the timing complexity of classical polynomial multiplication, Polynomial Substruction Polynomial addition, Karatsuba polynomial multiplication and Shuffling method, respectively. x represents the number of random shuffling operations.

Table 3. Comparison of polynomial multiplication techniques based random shuffling for BRLWE

Scheme	Computation time (ms)		Energy consumption (mJ)		Memory usage (bytes)	
	Enc	Dec	Enc	Dec	Enc	Dec
SPMA-Sh for BRLWE	49.8	29.5	1.83	1.07	180	132
2-DKSh for BRLWE	29.3	14.4	1.05	0.51	180	120
3-DKSh for BRLWE*	18.97	9.53	0.7	0.35	168	132
4-DKSh for BRLWE	33.4	16.3	1.16	0.57	168	132
3-DMRKSh for BRLWE	25.8	12.7	1.41	0.69	160	128

The primary operations in our scheme include polynomial addition, subtraction, and multiplication, with multiplication being the most computationally intensive. Below is a detailed explanation of each operation's efficiency and its impact:

Polynomial Addition and Subtraction: These operations have a time complexity of O(n), where n is the degree of the polynomial. Due to their linear complexity, they contribute minimally to the overall computational load and thus have a limited impact on performance.

Standard Polynomial Multiplication: Standard polynomial multiplication has a time complexity of $O(n^2)$, While this method can be effective for small polynomials, it becomes computationally demanding as the polynomial degree increases, making it less suitable for larger polynomials, especially in resource-constrained IoT environments.

3-Decomposition Karatsuba Polynomial Multiplication: To address the limitations of standard multiplication, we employ 3-Decomposition Karatsuba polynomial multiplication, which reduces the computational cost from $O(n^2)$ to $O(n^{\log_3 6})$, which is approximately $O(n^{1.464})$ [20, 22]. This reduction represents a significant improvement over the standard Karatsuba algorithm (2-decomposition) with complexity $O(n^{1.585})$.

Shuffling Method: The shuffling method is used to enhance security by obfuscating data, but it has a relatively low impact on overall time complexity compared to polynomial multiplication. This method operates in linear time, adding minimal computational overhead.

6. Implementation and experimental results

In this section, we provide device-specific optimization of the proposed method. To emphasize the relevance of the scheme for the Internet of Things, we choose a widely used and resource-constrained microcontroller as the target platform for our implementation, namely the ARM Cortex-M0. To assess the effectiveness of our approach, we conducted a comprehensive comparison against the original BRLWE [9] algorithm and the enhanced versions [17, 21] of the BRLWE that are implemented on the same platform.

The ARM Cortex-M0 is a 32-bit microcontroller processor core designed for low-power, cost-sensitive, and resource-constrained applications. It offers a compact and efficient architecture, making it ideal for various embedded systems,

IoT devices, and other applications that require energy efficiency and real-time processing capabilities, such as:

Healthcare: Wearable health monitoring devices, such as heart rate monitors and fitness trackers, which require low power consumption and efficient processing.

Smart Home: Home automation sensors, such as temperature, humidity and motion sensors, operate on limited computational resources while ensuring reliable data transmission.

Industrial IoT (IIoT): Environmental monitoring sensors used in factories and warehouses to track parameters like air quality and machine vibration, often deployed in large numbers and optimized for energy efficiency.

Agriculture: Soil moisture sensors and environmental monitoring devices in precision agriculture, which rely on battery-powered devices, must be highly energy-efficient and reliable.

To benefit from the advantage of the 32-bit architecture, we applied the same vectorizing method proposed by [9] to optimize the polynomial multiplication in our Karatsuba multiplication implementation. This method enables storing two coefficients into one data word, thereby reducing the time required for each multiplication operation by 50%.

We implemented the proposed 3DKSh-BRLWE cryptographic scheme on a resource-constrained, 32-bit ARM Cortex M0 microcontroller, specifically using the Arduino Zero platform and its integrated development environment (IDE).

This setup enabled us to evaluate the efficiency of our scheme in terms of computation time, memory usage, and energy consumption—three critical metrics for Internet of Things (IoT) applications where resources are highly limited.

To assess the benefits of our approach, we implemented the baseline BRLWE with a proposed random shuffling algorithm using standard polynomial multiplication (referred to as SPMA-Sh for BRLWE). We then extended this with several variations of Karatsuba polynomial multiplication techniques to improve computational efficiency, including: 2-Decomposition Karatsuba-based shuffling for BRLWE, (2-DKSh), 3-Decomposition Karatsuba-based shuffling for BRLWE (3-DKSh), 4-Decomposition Karatsuba-based shuffling for BRLWE (4-DKSh), and 3-Decomposition Karatsuba with integrated modular reduction-based shuffling for BRLWE (3-DMRKSh). Each variation allowed us to investigate the trade-offs between computational cost, memory usage, and energy consumption, with particular attention to the suitability of each approach for the ARM Cortex M0 microcontroller.

The primary goal of introducing Karatsuba decomposition methods was to reduce the time complexity of polynomial multiplication, which is the most computationally demanding operation in BRLWE. As summarized in Table 3, the results demonstrate a clear advantage of using Karatsuba-based methods over standard polynomial multiplication. Among the variations, the 3DKSh-BRLWE approach demonstrated the best computation times, requiring only 18.97 ms for encryption and 9.53 ms for decryption. This represents a significant improvement over the SPMA-Sh method and even outperforms other Karatsuba decomposition methods. The 2-DKSh for BRLWE requires 23.9 ms for encryption and 14.4 ms for decryption. While this approach achieves faster computation than SPMA-Sh due to fewer polynomial multiplications, it is less efficient than 3DKSh-BRLWE because the 2-Decomposition structure has a lower degree of decomposition, resulting in a higher number of coefficient multiplications. Conversely, 4-DKSh for BRLWE requires 33.4 ms for encryption and 16.3 ms for decryption. Although it has a higher decomposition factor, the 4-Decomposition method performs more addition operations and intermediate calculations, which exceed the processing capabilities of the ARM Cortex M0. This limits its efficiency on this microcontroller and makes it less suitable for resource-constrained environments. Meanwhile, 3-DMRKSh for BRLWE requires 25.8 ms for encryption and 12.7 ms for decryption. While modular reduction in this variation reduces memory usage (discussed below), the additional operations required for integrated modular reduction increase computation time, making it less efficient than standard 3-Decomposition (3DKSh-BRLWE).

Memory usage is another critical factor for IoT devices with limited storage, such as the ARM Cortex M0. As shown in Table 3, each variation of the scheme impacted memory usage differently, based on the decomposition method and the addition of modular reduction. The SPMA-Sh for BRLWE consumes 180 bytes for encryption and 132 bytes for decryption. By contrast, 2-DKSh for BRLWE requires 180 bytes for encryption but reduces the decryption memory to 120 bytes, achieving modest memory savings during decryption. In 3-DKSh for BRLWE, the scheme uses 168 bytes for encryption and 132 bytes for decryption, offering a small reduction in encryption memory usage compared to SPMA-Sh while maintaining similar decryption memory requirements. In 4-DKSh for BRLWE, the memory requirements are 168 bytes for encryption and 132 bytes for decryption, similar to 3-DKSh, although it incurs higher computation costs. The 3-DMRKSh for BRLWE further reduces memory usage to 160 bytes for encryption and 128 bytes for decryption. This approach provides the most memory-efficient solution, achieving a 4.8% reduction in encryption memory and a 3% reduction in decryption memory compared to standard 3-DKSh for BRLWE. However, as noted earlier, this memory savings is accompanied by increased computation time due to additional modular reduction operations.

Energy consumption is also a key metric for IoT devices operating on limited power sources, such as batteries. Energy usage is determined by the power required for each operation in combination with the overall computation time. In our analysis, we observed that the power needed for 2-DKSh, 3-DKSh, and for 4-DKSh schemes for BRLWE is consistent at approximately 35 W. However, the 3-DMRKSh for BRLWE approach, which integrates modular reduction, has a higher power consumption of approximately 54 W. This increased power demand results from the additional operations required for modular reduction, which, when combined with the extended computation time, significantly increases energy consumption.

Overall, our results indicate that 3DKSh-BRLWE offers the best balance of low computation time, manageable memory usage, and energy efficiency, making it the optimal choice among the evaluated approaches for resource-constrained IoT applications. While the 3-DMRKSh for the BRLWE variant achieves minor memory savings, its increased computation and energy costs make it less favorable for devices with strict power and performance limitations.

To demonstrate the effectiveness of our proposed 3DKSh-BRLWE scheme, we conducted a comparative analysis of computation time in both the encryption and decryption phases against the original BRLWE scheme [9], as illustrated in Figure 6. The results indicate that 3DKSh-BRLWE achieves a 32.49% reduction in computation time for encryption and a 20.58% reduction for decryption compared to the original BRLWE scheme. This substantial improvement in computation efficiency highlights the advantage of using our 3-Decomposition Karatsuba approach for lightweight encryption schemes. The significant reduction in computation time, combined with the enhanced security features of our approach, positions 3DKSh-BRLWE as a highly promising solution for securing resource-constrained applications, especially IoT devices where computational resources are limited, and efficiency is crucial.

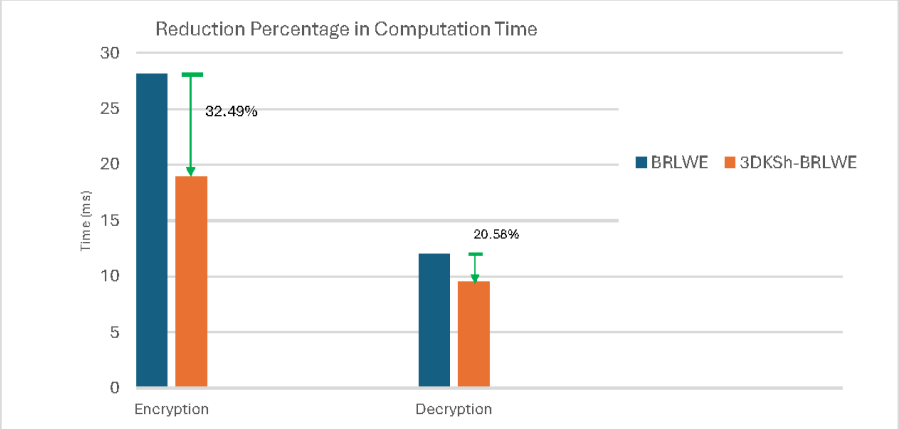

Figure 6. The reduction percentage of the original BRLWE and the proposed 3DKSh-BRLWE in the computation time

Additionally, we conducted a comparative study against the schemes presented in [17, 21], which were implemented using the same ARM Cortex M0 microcontroller and aimed to enhance BRLWE security. Our findings, summarized in Table 4, reveal that 3DKSh-BRLWE achieves the lowest computation time among the schemes tested, while also withstanding timing attacks, Simple Power Analysis (SPA), and Differential Power Analysis (DPA) attacks. By demonstrating resilience against these specific types of side-channel attacks, our proposed scheme achieves a higher level of security compared to the schemes in [17, 21], which are more susceptible to timing and power-based attacks due to longer computation times and less efficient shuffling methods.

Table 4. Implementation results compared with previous software implementations in arm Cortex-M0

Scheme	Parameters set	Resistance			Freq. MHz	Clock cycle (1K)		Time (ms)	
		Timing	SPA	DPA		Enc	Dec	Enc	Dec
InvRBLWE [17]	(p = 256, q = 256)	✓	×	×	–	1755	3404	52.3	101.4
Ring-ExpLWE [21]	(p = 256, q = 256)	✓	✓	×	32	1567	792	49.0	24.7
This research work * 3DKSh-BRLWE	(p = 256, q = 256)	✓	✓	✓	32	621	307	18.97	9.53

Overall, these results confirm that 3DKSh-BRLWE offers a unique combination of low computation time and robust security against side-channel attacks, providing a clear advantage over alternative schemes on the same ARM Cortex M0 platform. This makes it particularly well-suited for secure IoT applications, where efficient and resilient encryption is critical.

7. Conclusions

In this paper, we proposed an improved version of BRLWE using the random shuffling method and the 3-Decomposition Karatsuba multiplication to improve the security and optimize the polynomial multiplication cost.

We formally proved that 3DKSh-BRLWE resists the hybrid quantum attacks and the timing, SPA and DPA Side-Channel attacks. Moreover, we provide a software implementation using ARM Cortex-M0. In our evaluation, we examined the SPMA-based random shuffling for BRLWE with various Karatsuba multiplication techniques to demonstrate the advantages of Karatsuba multiplication and to determine the most suitable Karatsuba variant for our implementation. Furthermore, we compared our results with the original BRLWE and the existing software implementations that aim to improve it. Our approach requires only 18.79 ms and 9.53 ms for the encryption and the decryption respectively, which makes it a very good candidate for lightweight and resource-constrained IoT devices.

As future work, we aim to enhance the proposed approach to resist other types of side-channel attacks (SCAs), such as fault attacks. Our algorithm is currently optimized for polynomials of degree 256 and below on the Cortex M0, making it suitable for lightweight applications. However, for higher polynomial degrees, scalability needs to account for the increased cost of the random shuffling phase. To address this, implementing an adapted j-decomposition Karatsuba algorithm on more capable microcontrollers could help reduce computational overhead while maintaining efficiency in larger and more demanding cryptographic scenarios.

Acknowledgment

The authors would like to thank Dr. Franjieh El Khoury for her valuable comments and proofreading of this paper.

Funding

This research work was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), Prompt, Flex Group, and ISAME.

Conflict of interests

There is no conflict of interest declared by the authors.

References

[1] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, "Vision and challenges for realising the Internet of Things," in *Cluster of European Research Projects on the Internet of Things*, Brussels, Belgium: European Commission, vol. 3, no. 3, pp. 34–36, 2010.

[2] A. Deshpande, M. S. Deshpande, and D. N. Kayatanavar, "FPGA implementation of AES encryption and decryption," in *Porc. 2009 Int. Conf. Control, Autom., Commun. Energy Conserv.*, Perundurai, India, Jun. 4–6, 2009, pp. 1–6.

[3] X. Zhou, X. Tang, "Research and implementation of RSA algorithm for encryption and decryption," IEEE Xplore, Aug. 1, 2011. Accessed: May 28, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6021216.

[4] H. Marzouqi, M. Al-Qutayri, and K. Salah, "Review of Elliptic Curve Cryptography processor designs," *Microprocess. Microsyst.*, vol. 39, no. 2, pp. 97–112, 2015, https://doi.org/10.1016/j.micpro.2015.02.003.

[5] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, Santa Fe, NM, USA, Nov. 20–22, 1994, https://doi.org/10.1109/sfcs.1994.365700.

[6] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, 1997, https://doi.org/10.1137/s0097539795293172.

[7] L. Chen, L. Chen, S. Jordan, Y. K. Liu, D. Moody, R. Peralta, et al., *Report on Post-Quantum Cryptography*. Gaithersburg, MD, USA: US Department of Commerce, National Institute of Standards and Technology, vol. 12, 2016. https://doi.org/10.6028/nist.ir.8105.

[8] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, pp. 1–40, 2009, https://doi.org/10.1145/1568318.1568324.

[9] J. Buchmann, F. Göpfert, T. Güneysu, T. Oder, and T. Pöppelmann, "High-Performance and Lightweight Lattice-Based Public-Key Encryption," in *Proc. 2nd ACM Int. Workshop IoT Priv. Trust Security*, Xi'an, China, May 30, 2016, https://doi.org/10.1145/2899007.2899011.

[10] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *J. ACM*, vol. 60, pp. 1–35, 2013.

[11] R. Chaudhary, G. S. Aujla, N. Kumar, and S. Zeadally, "Lattice based Public Key Cryptosystem for Internet of Things Environment: Challenges and Solutions," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4897–4909, 2018, https://doi.org/10.1109/jiot.2018.2878707.

[12] S. Ebrahimi, S. Bayat-Sarmadi, and H. Mosanaei-Boorani, "Post-Quantum Cryptoprocessors Optimized for Edge and Resource-Constrained Devices in IoT," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5500–5507, 2019, https://doi.org/10.1109/jiot.2019.2903082.

[13] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem," in *STOC '09: Proc. 41st Annu. ACM Symp. Theory Comput.*, Bethesda, MD, USA, May 31–Jun. 2, 2009, https://doi.org/10.1145/1536414.1536461.

[14] A. Aysu, M. Orshansky, M. Tiwari, "Binary Ring-LWE hardware with power side-channel countermeasures," in *Proc. 2018 Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Dresden, Germany, Mar. 19–23, 2018, pp. 1253–1258, https://doi.org/10.23919/date.2018.8342207.

[15] P. C. Kocher, Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Proc. Adv. Cryptol.—CRYPTO'96: 16th Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, Aug. 18–22, 1996, pp. 104–113.

[16] T. Schneider, A. Moradi, and T. Güneysu, "ParTI—Towards Combined Hardware Countermeasures against Side-Channel and Fault-Injection Attacks," *IACR Cryptol. ePrint Archive*, vol. 2016, p. 648, 2016.

[17] S. Ebrahimi and S. Bayat-Sarmadi, "Lightweight and Fault Resilient Implementations of Binary Ring-LWE for IoT Devices," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6970–6978, 2020, https://doi.org/10.1109/jiot.2020.2979318.

[18] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, "Introduction to differential power analysis," *J. Cryptographic Eng.*, vol. 1, no. 1, pp. 5–27, Mar. 2011, https://doi.org/10.1007/s13389-011-0006-y.

[19] O. Reparaz, S. Sinha Roy, F. Vercauteren, I. Verbauwhede, "A Masked Ring-LWE Implementation," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Berlin, Heidelberg: Springer, 2015, pp. 683–702. https://doi.org/10.1007/978-3-662-48324-4_34.

[20] A. A. Karatsuba, Y. P. Ofman, "Multiplication of many-digital numbers by automatic computers," *Dokl. Akad. Nauk SSSR*, vol. 145, no. 2, pp. 293–294, 1962. Accessed: Aug. 9, 2023. [Online]. Available: https://www.mathnet.ru/eng/dan26729.

[21] D. Xu, X. Wang, Y. Hao, Z. Zhang, Q. Hao, H. Jia, et al., "Ring-ExpLWE: A High-performance and Lightweight Post-quantum Encryption Scheme for Resource-constrained IoT Devices," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24122–24134, 2022, https://doi.org/10.1109/jiot.2022.3189210.

[22] P. He, Y. Tu, J. Xie, H. S. Jacinto, "KINA: Karatsuba Initiated Novel Accelerator for Ring-Binary-LWE (RBLWE)-based Post-Quantum Cryptography," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 31, no. 10, pp. 1551–1564, Aug. 2022.

[23] "Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process," Accessed: Aug. 9, 2023. [Online]. Available: https://csrc.nist.gov/Projects/post-quantumcryptography/publications.

[24] M.-S. Chen and T. Chou, "Classic McEliece on the ARM Cortex-M4," *IACR Trans. Cryptogr. Hardw. Embedded Syst.*, vol. 2021, pp. 125–148, Jul. 2021, https://doi.org/10.46586/tches.v2021.i3.125-148.

[25] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, et al., "CRYSTALS—Kyber: A CCA-Secure Module-Lattice-Based KEM," in *Proc. 2018 IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, London, UK, Apr. 24–26, 2018, https://doi.org/10.1109/eurosp.2018.00032.

[26] M. R. Albrecht, B. R. Curtis, A. Deo, A. Davidson, R. Player, E. W. Postlethwaite, et al., "Estimate all the LWE, NTRU schemes!" *IACR Cryptol. ePrint Archive*, vol. 2018, p. 331, Jan. 2018.

[27] M. V. Beirendonck, J.-P. D'anvers, A. Karmakar, J. Balasch, and I. Verbauwhede, "A Side-Channel-Resistant Implementation of SABER," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 17, no. 2, pp. 1–26, Apr. 2021, https://doi.org/10.1145/3429983.

[28] D. Micciancio, "The hardness of the closest vector problem with preprocessing," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 1212–1215, Mar. 2001, https://doi.org/10.1109/18.915688.

[29] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem: Extended abstract," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, Bethesda, MD, USA, May 31–Jun. 2, 2009, pp. 333–342.

[30] J. Richter-Brockmann, J. Mono, and T. Guneysu, "Folding BIKE: Scalable Hardware Implementation for Reconfigurable Devices," *IEEE Trans. Comput.*, vol. 71, no. 5, pp. 1204–1215, May 2022, https://doi.org/10.1109/tc.2021.3078294.

[31] J. Howe, T. Oder, M. Krausz, and T. Güneysu, "Standard Lattice-Based Key Encapsulation on Embedded Devices," *IACR Trans. Cryptographic Hardware Embedded Syst.*, pp. 372–393, Aug. 2018, https://doi.org/10.46586/tches.v2018.i3.372-393.

[32] H. Seo, M. Anastasova, A. Jalali, and R. Azarderakhsh, "Supersingular Isogeny Key Encapsulation (SIKE) Round 2 on ARM Cortex-M4," *IEEE Trans. Comput.*, vol. 70, no. 10, pp. 1705–1718, Oct. 2021, https://doi.org/10.1109/tc.2020.3023045.

[33] X. Qian and W. Wu, "An Efficient Ciphertext Policy Attribute-Based Encryption Scheme from Lattices and Its Implementation," in *Proc. 2021 IEEE 6th Int. Conf. Comput. Commun. Syst. (ICCCS)*, Chengdu, China, Apr. 23–26, 2021, https://doi.org/10.1109/icccs52626.2021.9449182.

[34] I. Mustafa, I. U. Khan, S. Aslam, A. Sajid, S. M. Mohsin, M. Awais, et al., "A Lightweight Post-Quantum Lattice-Based RSA for Secure Communications," *IEEE Access*, vol. 8, pp. 99273–99285, 2020, https://doi.org/10.1109/access.2020.2995801.

[35] A. Abdulrahman, J.-P. Chen, Y.-J. Chen, V. Hwang, M. J. Kannwischer, and B.-Y. Yang, "Multi-moduli NTTs for Saber on Cortex-M3 and Cortex M4," *IACR Trans. Cryptogr. Hardw. Embedded Syst.*, vol. 2021, pp. 127–151, Nov. 2021, https://doi.org/10.46586/tches.v2022.i1.127-151.

[36] C. M. M. Chung, V. Hwang, M. J. Kannwischer, G. Seiler, C. J. Shih, B. Y. Yang, "NTT multiplication for NTT-unfriendly rings: New speed records for Saber and NTRU on Cortex-M4 and AVX2," *IACR Trans. Cryptogr. Hardw. Embedded Syst.*, pp. 159–188, Feb. 2021, https://doi.org/10.46586/tches.v2021.i2.159-188.

[37] Z. Y. Wong, D. C. K. Wong, W. K. Lee, K. M. Mok, W. S. Yap, A. Khalid, "KaratSaber: New Speed Records for Saber Polynomial Multiplication using Efficient Karatsuba FPGA Architecture," *IEEE Trans. Comput.*, vol. 72, no. 7, pp. 1830–1842, Jan. 2023, https://doi.org/10.1109/tc.2023.3238129.

[38] A. Sarker, M. M. Kermani, and R. Azarderakhsh, "Error Detection Architectures for Ring Polynomial Multiplication and Modular Reduction of Ring-LWE in $\frac{\mathbb{Z}/p\mathbb{Z}[x]}{x^n+1}$ Benchmarked on ASIC," *IEEE Trans. Reliab.*, vol. 70, no. 1, pp. 362–370, Mar. 2021, https://doi.org/10.1109/tr.2020.2991671.

[39] A. Khalid, T. Oder, F. A. Valencia, M. O'Neill, T. Güneysu, and F. Regazzoni, "Physical Protection of Lattice-Based Cryptography," in *GLSVLSI '18: Proc. 2018 Great Lakes Symp. VLSI*, Chicago, IL, USA, May 23–25, 2018, https://doi.org/10.1145/3194554.3194616.

[40] J. R. Howe, A. Khalid, C. Rafferty, F. Regazzoni, and M. O'Neill, "On Practical Discrete Gaussian Samplers for Lattice-Based Cryptography," *IEEE Trans. Comput.*, vol. 67, no. 3, pp. 322–334, Mar. 2018, https://doi.org/10.1109/tc.2016.2642962.

[41] S. Ebrahimi, S. Bayat-Sarmadi, "Lightweight and DPA-Resistant Post-Quantum Cryptoprocessor based on Binary Ring-LWE," in *Proc. 2020 20th Int. Symp. Comput. Arch. Digit. Syst. (CADS)*, Rasht, Iran, Aug. 19–20, 2020, https://doi.org/10.1109/cads50570.2020.9211858.

[42] S. Ahmadunnisa, S. E. Mathe, "CNC: A lightweight architecture for Binary Ring-LWE based PQC," *Microprocess. Microsyst.*, vol. 106, p. 105044, Apr. 2024, https://doi.org/10.1016/j.micpro.2024.105044.

[43] X. Zhang, Z. Huai, and K. K. Parhi, "Polynomial Multiplication Architecture with Integrated Modular Reduction for R-LWE Cryptosystems," *J. Signal Process. Syst.*, vol. 94, no. 8, pp. 799–809, Apr. 2022, https://doi.org/10.1007/s11265-022-01746-7.

[44] X. Zhang and K. K. Parhi, "Reduced-Complexity Modular Polynomial Multiplication for R-LWE Cryptosystems," in *Proc. ICASSP 2021—2021 IEEE Int. Conf. Acoust. Speech Signal Process. Process. (ICASSP)*, Toronto, ON, Canada, Jun. 6–11, 2021, pp. 7853–7857, https://doi.org/10.1109/ICASSP39728.2021.9414005.

[45] D. E. Knuth, *The Art of Computer Programming*, vol. 2, 3rd Ed. Seminumerical Algorithms. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1997.

[46] P. Pessl, "Analyzing the Shuffling Side-Channel Countermeasure for Lattice-Based Signatures," in *Proc. Cryptol.—INDOCRYPT 2016: 17th Int. Conf. Cryptol. India*, Kolkata, India, Dec. 11–14, 2016, pp. 153–170, https://doi.org/10.1007/978-3-319-49890-4_9.

[47] T. Wunderer, "Revisiting the Hybrid Attack: Improved Analysis and Refined Security Estimates," Cryptology ePrint Archive, 2016. Accessed: Mar. 8, 2023. [Online]. Available: https://eprint.iacr.org/2016/733.

[48] F. Göpfert, C. van Vredendaal, and T. Wunderer, "A hybrid lattice basis reduction and quantum search attack on LWE," in *Post-Quantum Cryptogr.: 8th Int. Workshop, PQCrypto 2017*, Utrecht, The Netherlands, Jun. 26–28, 2017, pp. 184–202, https://doi.org/10.1007/978-3-319-59879-6_11.

[49] L. Babai, "On Lovász' lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986, https://doi.org/10.1007/bf02579403.