Post-Quantum Cryptoprocessors Optimized for Edge and Resource-Constrained Devices in IoT

Shahriar Ebrahimi, Siavash Bayat-Sarmadi, *Member, IEEE*, Hatameh Mosanaei-Boorani

***Abstract*—By exponential increase in applications of the internet of things (IoT), such as smart ecosystems or e-health, more security threats have been introduced. In order to resist known attacks for IoT networks, multiple security protocols must be established among nodes. Thus, IoT devices are required to execute various cryptographic operations such as public key encryption/decryption. However, classic public key cryptosystems such as RSA and ECC are computationally more complex to be efficiently implemented on IoT devices and are vulnerable regarding quantum attacks. Therefore, after complete development of quantum computing, these cryptosystems will not be secure and practical. In this paper, we propose InvRBLWE, an optimized variant for binary learning with errors over the ring (Ring-LWE) scheme that is proven to be secure against quantum attacks and is highly efficient for hardware implementations. We propose two architectures for InvRBLWE: 1) a high-speed architecture targeting edge and powerful IoT devices, 2) an ultra-lightweight architecture, which can be implemented on resource-constrained nodes in IoT. The proposed architectures are scalable regarding security levels and we provide experimental results for two versions of the InvRBLWE scheme providing 84 and 190 bits of classic security. Our implementation results on FPGA dominate the best of the classic and post-quantum previous implementations. Moreover, our two different ASIC implementations show improvement in terms of speed, area, power and/or energy. To the best of our knowledge, we are the first to implement LWE-based cryptosystems on ASIC platform.**

***Index Terms*—Post-quantum cryptography, lattice-based cryptography, ring learning with errors (Ring-LWE), internet of things (IoT), hardware implementation.**

I. Introduction

Internet of Things (IoT) introduces a dynamic and highly adaptive network by extending the connectivity abilities to embedded devices such as sensors and actuators. Therefore, in recent years, IoT applications such as smart transportation, home automation, manufacturing automation and e-healthcare have grown exponentially among various customers ranging from end-users to big organizations. Connecting more IoT devices, especially resource-constrained ones, to the network results in higher security threats [1]. While IoT shapes the future of internet, communications among IoT nodes must be secured against different attacks, malwares and viruses. To implement secure channels for communications, each IoT node must execute at least a few operations from cryptographic primitives such as public key encryption/decryption. Fig. 1 presents the overall architecture of current advanced IoT networks that are constructed from three major layers: 1) cloud, 2) edge devices and 3) IoT end-nodes. While servers and some of edge devices can benefit from high-performance resources such as 64-bit processors and field programmable gate arrays (FPGA), the end-nodes have constrained resources. The practical cryptographic hardware solution for such resource-constrained devices can only be achieved through implementation of the architecture on application specific integrated circuit (ASIC) platforms.

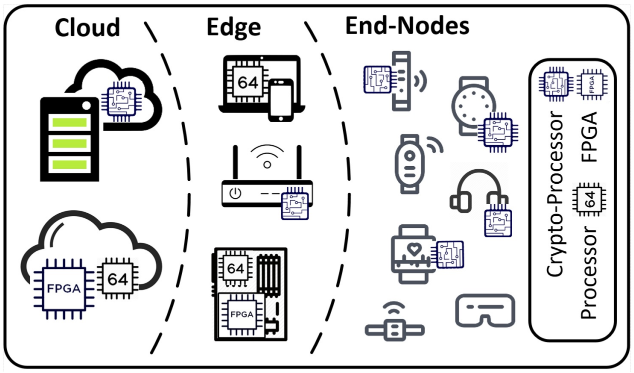

Fig. 1. Architecture of IoT Network and Available Hardware Resources in each Layer.

We note that most of classic public key cryptosystems are computationally more complex to be efficiently implemented on IoT nodes [2], [3]. In addition to such high complexity, current classic public key cryptosystems such as Rivest-Shammir-Adlemon (RSA) [4] and elliptic curve cryptography (ECC) [5], [6] rely on hard problems that have polynomial time solutions using quantum search algorithm [7]. Thus, when quantum computing gains required computation power, all of classic cryptosystems loose their security levels and will need very large keys to remain secure. This makes current implementations of RSA and ECC impractical in post-quantum era due to increased hardware implementation complexity. Therefore, it is necessary to consider alternative cryptosystems that rely on other hard problems and are quantum-resistant. Some examples include shortest/closest vector problems (SVP/CVP) in lattices [8], learning with errors (LWE) [9] and its variants [10]–[12], code-based cryptography [13] and isogenies over elliptic curve [14], [15]. There are multiple cryptosystems proposed based on aforementioned hard problems among which lattice-based cryptography is more practical for resource-constrained nodes in IoT due to its relatively fast operations with low complexity. In the recent post-quantum cryptography standardization by national

All authors are with the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. Emails: shebrahimi@ce.sharif.edu, sbayat@sharif.edu, mosanaei@ce.sharif.edu.

institute of standard and technology (NIST) [16], most of the proposed schemes in the first round of general submissions are based on lattice hard problems [17]–[19].

In 2009, Regev et al. introduced a new lattice-based cryptosystem that relies on the hardness of the LWE problem [9]. The search version of the problem is defined over finding secret s in a linear combination $a.s + e$, where a is known and e is an error according to a certain distribution. LWE and its variants have shown resistance to various types of classic and quantum attacks [10]–[12], [20]–[22]. In 2010, a new variant of LWE based on ring theory introduced in [10], called Ring-LWE. It utilizes ideal lattices and has relatively smaller key sizes compared to the original LWE scheme. In 2016, Buchmann et al. proposed a new variant of Ring-LWE, namely Ring-BinLWE, by choosing errors from a binary distribution instead of Gaussian one [11].

Many hardware implementations for LWE-based schemes have been proposed [17], [23]–[25]. Recently, a hardware implementation based on Ring-BinLWE [11] has been proposed in [25] that is relatively faster than previous implementations for Ring-LWE schemes.

In this paper, we propose a hardware-optimized variant of Ring-BinLWE, hereafter referred to as InvRBLWE. Our optimization utilizes inverted ring of Ring-BinLWE and therefore, matches 2's-complement notation range that is highly optimized for hardware implementation. In Section III, we justify that operations over the ring require no reduction when ring elements are presented in 2's-complement notation and thus, the entire reduction operations are omitted from the InvRBLWE scheme. We propose two architectures for InvRBLWE targeting different IoT devices:

1) **High-Speed:** a fast and low-complexity architecture, which is a good match for *edge* and *high-performance* devices in IoT.
2) **Ultra-Lightweight:** an ultra low-power and low area architecture targeting *resource-constrained* end-nodes that are powered by *batteries* or *energy harvesting* units in IoT.

Regarding implementation results, the high-speed architecture has been implemented on FPGA that is considered a high-performance platform practical on edge and powerful IoT devices. Our FPGA implementations dominate previous hardware implementations of any LWE-based scheme by improving $Area \times Time$ (AT) complexity with at least 52% in encryption/decryption operations. The proposed architectures for InvRBLWE are platform-independent and can also be implemented on ASIC platforms, which are ideal for crypto-processors in IoT devices. The ASIC high-speed implementation results appear to be at least two times faster than previous work. Additionally, the ultra-lightweight ASIC implementation consumes 62% and 66% less area and power compared to the most lightweight implementation of ECC, respectively. Furthermore, proposed architectures are shown to be resistant against simple power analysis (SPA) [26] and timing attacks [27]. The main contributions of this paper are summarized as follows:

- A variant of the Ring-BinLWE scheme has been proposed, namely InvRBLWE, which is fully optimized for hardware implementation.
- The operation cost in the proposed scheme is reduced by omitting all of the required reduction operations.
- Our high-speed implementation dominates previous work in terms of speed on both FPGA and ASIC platforms.
- The high-speed ASIC implementations of LWE-based cryptosystems requires significantly lower energy compared to the best AISC implementations of classic and post-quantum cryptosystems. This architecture best suits for battery-based IoT devices.
- The ultra-lightweight implementation of InvRBLWE on ASIC platform consumes less power and area compared to that ECC requires. To the best of our knowledge, this is the first public key implementation to have such low power consumption that can even be supplied by Vibration Piezo or Electro Magnetic (EM) [3] energy harvesting units.

The rest of the paper is organized as follows. In Section II, required background is described. Section IV provides detailed information of the proposed scheme and architectures. We compare our implementation results against previous work on both FPGA and ASIC platforms in Section V. Finally, the paper is concluded in Section VI.

II. Preliminaries

In this section, we describe the required background and notations to follow the rest of the paper. At first, we make a few clarifications regarding mathematical notations used in the paper. Later, the paper presents a brief history of developments in lattice-based cryptosystems. Finally, we describe the Ring-BinLWE scheme [11] in details.

A. Ring Theory

For an integer $q \in \mathbb{Z}$, we define the finite ring $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z} = \{0, 1, ..., q-1\}$. In other words, $\mathbb{Z}_q$ is a finite ring, where q is its modulus. For an integer $n \in \mathbb{Z}$, the set of all n-dimensional vectors, which each dimension belongs to $\mathbb{Z}_q$, is denoted as $\mathbb{Z}_q^n = \{\langle w_0, w_1, ..., w_{n-1}\rangle | w_i \in \mathbb{Z}_q\}$.

The set of all polynomials whose coefficients belong to $\mathbb{Z}$ is shown as $\mathbb{Z}[x]$. Similarly, we refer to the set of all polynomials that have their coefficients chosen from $\mathbb{Z}_q$, as $\mathbb{Z}_q[x]$. Thus, each vector in $\mathbb{Z}_q^n$ can be mapped to a unique polynomial of degree $n-1$ in $\mathbb{Z}[x]$. In addition to simple rings over integers, we can extend ring definitions to polynomials. Therefore, the ring of polynomials $\mathcal{R}$ is defined as $\mathcal{R} = \mathbb{Z}[x]/f(x)$, where $f(x)$ is the modulus of the ring. If all of the coefficients of the polynomials in the ring are chosen from $\mathbb{Z}_q$, we refer to the polynomial ring as $\mathcal{R}_q = \mathbb{Z}_q[x]/f(x)$.

B. Ring-BinLWE Scheme

In 2009, Regev et al. introduced a new lattice-based cryptosystem that relies on the hardness of the LWE problem [9]. The search version of the problem is defined as finding a secret $s \in \mathbb{Z}_q^n$ hidden in a known pair (a, b) such that $b = as + e$, while $b, a \in \mathbb{Z}_q^n$ and $e \in \mathbb{Z}_q$. Moreover, $a \in \mathbb{Z}_q^n$ is known and uniformly random and $e \in \mathbb{Z}_q$ is an error according to a

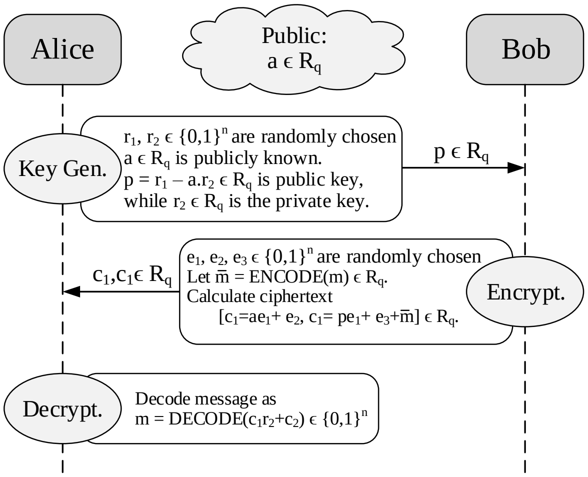
Alice
Public:
a ϵ R$_q$
Bob
Key Gen.
r$_1$, r$_2$ ϵ {0,1}^{n} are randomly chosen
a ϵ R$_q$ is publicly known.
p = r$_1$ – a.r$_2$ ϵ R$_q$ is public key,
while r$_2$ ϵ R$_q$ is the private key.
p ϵ R$_q$
e$_1$, e$_2$, e$_3$ ϵ {0,1}^{n} are randomly chosen
Let m̄ = ENCODE(m) ϵ R$_q$.
Calculate ciphertext
[c$_1$=ae$_1$+ e$_2$, c$_1$= pe$_1$+ e$_3$+m̄] ϵ R$_q$.
Encrypt.
c$_1$,c$_1$ϵ R$_q$
Decrypt.
Decode message as
m = DECODE(c$_1$r$_2$+c$_2$) ϵ {0,1}^{n}

Fig. 2. Ring-BinLWE Scheme.

distribution ψ over $\mathbb{Z}_q$. There are different choices proposed for ψ such as Gaussian [9] and binary [11], [12].

A more efficient and practical variant of the LWE scheme has been introduced in [10], called Ring-LWE that is defined over $\mathcal{R}_q = \mathbb{Z}_q[x]/f(x) = (-\lfloor \frac{q}{2} \rfloor + 1, \lfloor \frac{q}{2} \rfloor)$. There are multiple choices for $f(x)$ as the modulus for the ring $\mathcal{R}_q$. Assigning $f(x) = x^n + 1$ causes the shift operation of the ring to turn into an anti-circular rotation [10] that is more efficient for software and hardware implementation. It is worth mentioning that after each operation over the ring $\mathcal{R}_q$ the result must be checked to decide whether a reduction is required or not. The reduction module in both hardware increases overall area and critical path delay (CPD) of the system.

There are multiple variants of LWE proposed by researchers that suggest new choices for LWE characteristics [12], [20]. In 2016, Buchmann et al. proposed a variant of the Ring-LWE problem using binary error distribution [11]. Ring-BinLWE has smaller key sizes and does not require any complex operations such as convolutions in Gaussian distribution. Detailed information of the Ring-BinLWE scheme operations is provided in Fig. 2. The scheme consists of three main phases: 1) *Key Generation*, 2) *Encryption* and 3) *Decryption*. We describe the operations in each phase in the following:

- **Key Generation:** This phase starts with choosing two random binary vectors $r_1, r_2 \in \{0,1\}^n$. These vectors can be mapped to unique polynomials in $\mathcal{R}_q$. In addition, a publicly known polynomial $a \in \mathcal{R}_q$ is used to calculate the public key $p = r_1 - a.r_2 \in \mathcal{R}_q$. The binary vector r_1 is a one-time error and will be discarded afterwards, while r_2 is Alice's private key. In this scheme, the private and public keys consist of n and $n \times log_2^q$ bits, respectively.
- **Encryption:** To encrypt an n-bit message $m \in \{0,1\}^n$, Bob encodes the message as a unique polynomial $\bar{m}$ in R_q using the function described in Equation (1). Then three n-bit random binary vectors $e_1, e_2, e_3 \in \mathcal{R}_q$ are chosen to calculate the ciphertext using the known polynomial a and Alice's public key p. The ciphertext consists of a pair of two polynomials $c_1 = a.e_1 + e_2$ and $c_2 = p.e_1 + e_3 + \bar{m}$ in $\mathcal{R}_q$. The length of the ciphertext is $2 \times n \times log_2^q$ bits.

$$\boxed{\begin{array}{l} ENCODE : \{0,1\}^n \longrightarrow \mathcal{R}_q, \\ (m_0, ..., m_{n-1}) \longrightarrow \sum_{i=0}^{n-1} m_i (\frac{q}{2}) x^i. \end{array}} \tag{1}$$

- **Decryption:** To be able to decrypt the ciphertext, Alice uses the private key r_2 to calculate $\bar{m} = c_1.r_2 + c_2 \in \mathcal{R}_q$. Alice uses the function described in Equation (2) to decode $\bar{m}$.

$$\boxed{\begin{array}{l} DECODE : \mathcal{R}_q \longrightarrow \{0,1\}^n, \\ \sum_{i=0}^{n-1} a_i x^i \rightarrow (m_0, ..., m_{n-1}), \\ m_i = \begin{cases} 1 & |a_i - i - \lfloor \frac{n-3}{2} \rceil | > \frac{q}{4}, \\ 0 & \text{else.} \end{cases} \end{array}} \tag{2}$$

The main advantage of the Ring-BinLWE compared to the other Ring-LWE-based schemes is the binary error distribution, which can be implemented on hardware more efficiently. Ring-BinLWE improves key and ciphertext sizes compared to the standard Ring-LWE [11].

There are multiple classic and quantum attacks introduced against LWE problem with binary errors [11], [12], [22]. According to the latest and most successful attacks in [22], choosing parameter set of $n = 256$ and $q = 256$ provides 84 and 73 bits of classic and quantum security level, respectively. Moreover, the parameter set of $n = 512$ and $q = 256$ results in 190 and 140 bits of classic and quantum security level, respectively.

III. Scheme Optimization

As discussed in previous section, one of the critical modules in implementation of ring $\mathcal{R}_q$ operations is *reduction*, which should be evaluated after every single operation. In order to make Ring-BinLWE [11] more efficient for hardware implementation, we re-consider the underlying ring $\mathcal{R}_q$ and propose an optimized version of the scheme, called InvRBLWE. In the proposed InvRBLWE scheme, the coefficients of the polynomials in $\mathcal{R}_q$ are selected from the inverted range compared to the original Ring-BinLWE. In InvRBLWE, every member $A(x)$ in $\mathcal{R}_q$ is defined as $A(x) = \sum_{i=0}^{n-1} \alpha_i x^i$, where $\alpha_i \in \mathbb{Z}_q$, which in this scheme is represented in the range of $(-\lfloor \frac{q}{2} \rfloor, \lfloor \frac{q}{2} \rfloor - 1)$; notice the difference with the original range of Ring-LWE. In InvRBLWE, the range of coefficients, exactly matches the 2's-complement notation range for a $\log_2^q$-bit integer. The advantage of InvRBLWE is that the reductions over the inverted ring can be handled easily in hardware implementation. We show that all necessary reductions for any modular operation are performed automatically by normal overflow and underflow in 2's-complement notation according to Lemma 1.

Lemma 1. *Assume $q = 2^k$ and $k \in \mathbb{Z}$. Moreover, the ring $\mathbb{Z}_q$ is represented with 2's-complement notation. Then, modular addition/subtraction of two members $a, b \in \mathbb{Z}_q$ does not require any reduction.*

Proof. The opposite of a number x in 2's-complement notation is computed as $\bar{x} = 2^k - x$. As mentioned earlier, the underflow/overflow in 2's-complement notation exactly matches underflow/overflow in ring $\mathbb{Z}_q$ operations. In case of overflow, in order to execute a reduction in ring $\mathbb{Z}_q$, we clearly subtract q from the result. The Equation (3) computes the reduction in this scenario. Similarly, for the case of underflow, an addition by q has to be performed. This is shown in Equation (4).

REDUCTION: If $x \geq \frac{q}{2}$: then

$$\begin{aligned} x_{new} &= x - q = x + \bar{q} = x + (2^k - q) \\ &= x + (2^k - 2^k) = x + 0 \implies x_{new} = x. \end{aligned} \tag{3}$$

REDUCTION: If $x < -\frac{q}{2}$: then

$$\begin{aligned} x_{new} &= x + q = x + 2^k \\ &= 2^k - \bar{x} = -\bar{x} \implies x_{new} = x. \end{aligned} \tag{4}$$

Thus, no reduction is required once elements of $\mathbb{Z}_q$ are presented using 2's-complement notation with $k = \log_2^q$ bits. □

Despite the notation of the underlying ring, the entire operations and parameters of InvRBLWE are exactly same as the original Ring-BinLWE. Hence, all of the correctness and security claims in [12] and [11] also apply to InvRBLWE. In the proposed InvRBLWE, all of the modular operations are performed in an inverted manner compared to the standard Ring-BinLWE. In order to maintain the correctness of the scheme, the encode and decode functions have to be updated accordingly. The encode function in original Ring-BinLWE, maps a binary vector $m \in \{0,1\}^n$ to a vector $\bar{m} \in \mathcal{R}_q$ by assigning $\frac{q}{2}$ to every none-zero bits in the message [11]. In the notation used in InvRBLWE, which is same as 2's-complement, $\frac{q}{2}$ is congruent to $-\frac{q}{2}$. Thus, in contrast to the original encode function [11], the proposed encode function for InvRBLWE maps every none-zero bits of the message m to $-\frac{q}{2}$ as shown in Equation (5). The decode function is similarly updated to be the opposite of the original decode function presented in [11], as shown in Equation (6).

$$\boxed{\begin{aligned} &ENCODE : \{0,1\}^n \longrightarrow \mathcal{R}_q, \\ &(m_0, ..., m_{n-1}) \longrightarrow \textstyle\sum_{i=0}^{n-1} m_i(-\frac{q}{2})x^i. \end{aligned}} \tag{5}$$

$$\boxed{\begin{aligned} &DECODE : \mathcal{R}_q \longrightarrow \{0,1\}^n, \\ &\textstyle\sum_{i=0}^{n-1} a_i x^i \rightarrow (m_0, ..., m_{n-1}), \\ &m_i = \begin{cases} 0 & |a_i - i - \lfloor \frac{n-3}{2} \rceil |> \dfrac{q}{4}, \\ 1 & \text{else.} \end{cases} \end{aligned}} \tag{6}$$

We note that every implementation over the proposed InvRBLWE scheme, can benefit from omission of reduction modules compared to the standard scheme that has been implemented in [25] and [11]. Therefore, as we show in Section IV and Section V, our implementations of binary Ring-LWE have lower complexity, which result in faster operations and higher efficiency compared to previous work.

IV. Proposed Architecture

In this section, we discuss the proposed architecture and its detailed features. First, we provide information for our high-speed architecture, which offers straightforward solution regarding implementation InvRBLWE scheme. Later, we optimize the area of the proposed scheme for tiny devices to achieve an ultra-lightweight architecture for InvRBLWE. Finally, the proposed architectures are analyzed for resistance against SPA and timing attacks.

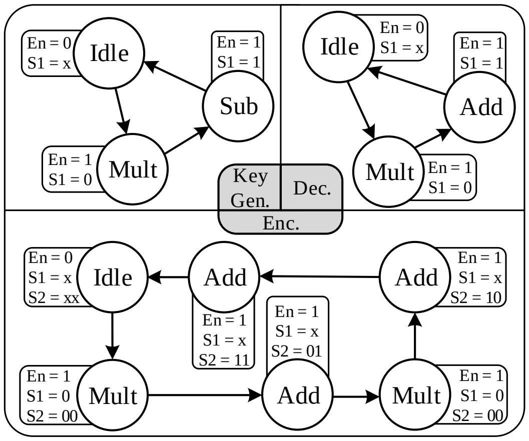

Fig. 3. FSM Diagram for the InvRBLWE Scheme.

A. High-Speed Architecture

As shown in Section III, the InvRBLWE scheme can be optimized for hardware implementation due to the fact that it does not require any reduction over the ring $\mathcal{R}_q$ (while $q = 2^k$). Thus, unlike the standard Ring-BinLWE implementations [25], all of the modular operations can be implemented without any considerations for reduction. Therefore, while decreasing the CPD, the required hardware contains only n parallel adders with a few control signals. Different choices for scheme parameters (n and q) provide multiple security levels of Ring-BinLWE [11], [12]. In this paper, we have chosen two parameter sets of Ring-BinLWE to be implemented by the InvRBLWE scheme. The scheme parameter q is set to 256, while n is set to 256 and 512 to achieve 84 and 190 bits of security, respectively [11].

We propose an all-in-one unified architecture for all three main phases of the scheme (key generation, encryption and decryption) for hardware implementation using scheme parameter q = 256. Fig. 3 shows the finite state machine (FSM) of all three scheme operations. As shown in Fig. 2, encryption has the highest time-complexity because of its two multiplication and three addition operations. In the following, the detailed information regarding hardware implementation of decryption phase is provided. The other two phases (key generation and encryption) have similar types of operations as shown in Fig. 2 and Fig. 3 and their implementations have only a few alternative control signals. Because of common use of entire registers and adder units between different phases of the scheme, all of the three phases are implemented using one unified architecture.

Decryption: Fig. 4 shows proposed architecture for decryption. In this phase, one must calculate $\bar{m} = c_1 + c_2.r_2$ that consists of one multiplication and one addition. The architecture must decode the calculated $\bar{m}$ using function provided in Equation (6). As shown in Fig. 3, during the multiplication, the control signal S_1 is set to zero. Therefore, all of the

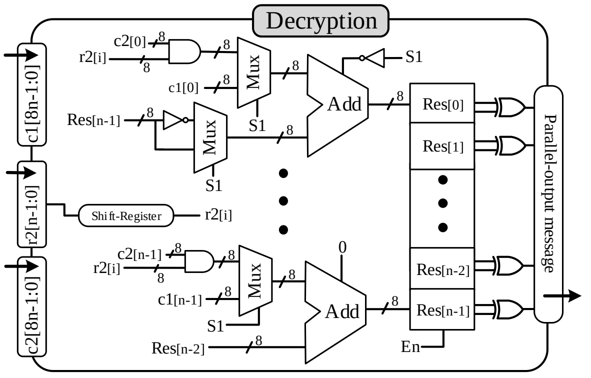

Fig. 4. High-Speed Architecture for Decryption Phase.

adders, except the first one, are performing add function. The multiplication inputs are $a \in \mathcal{R}_q$ and $r_2 \in \{0,1\}^n$. Hence, the multiplication can be calculated in n clock cycles using the shift-and-add method, which requires n parallel adders of 8-bit ($\log_2^q$-bit) length as shown in Fig. 4. It is worth mentioning that r_2 is loaded into a shift-register, which outputs each bit of r_2 in certain clock cycle during multiplication operation.

Due to the characteristics of the ring $\mathcal{R}_q$, the shift operation is performed using an anti-circular rotation because of the modulus $f(x) = x^n + 1$ [11]. To implement such anti-circular rotation in hardware, we simply feed each 8-bit ($\log_2^q$-bit) register to the input of the next adder and finally feed the negative of Res[n-1] (the last 8-bit register) to the first adder. Setting carry_in of the first adder to 1, changes its function to subtraction, which completes the anti-circular rotation. During multiplication, control signal S_1 is set to zero and the first adder performs subtraction. The rest of the adders always perform addition and the corresponding carry_in signal is set to zero.

After completion of the multiplication, an additional clock cycle is required to calculate the final addition operation. In this state, the control signal S_1 is set to one and all of the adders perform addition. The public key p is available after $n + 1$ clock cycles. During each clock cycle, the results are stored in n registers of 8-bit length. It is worth mentioning that r_2 is a binary vector and each bit of r_2 needs to be extended to 8 sequential bits ($\log_2^q$-bit). This bit adjustment is performed in hardware with no significant overhead by simple rewiring. At the end of decryption phase, the result must be decoded from a polynomial in $\mathcal{R}_q$ to a binary vector based on Equation (6). We implement the decode function by an array of n 2-input XOR gates that compare two most significant bits of each register block, namely *Res[7]* and *Res[6]*.

B. Ultra-Lightweight Architecture

In order to achieve a lightweight architecture for InvRBLWE scheme, instead of the straightforward and parallel use of components (i.e., 8-bit adders and registers), we have utilized only one set of such components in a serial manner as shown in Fig. 5. Moreover, contrasting with the high-speed method to perform multiplication, the multiplicand is shifted during each cycle instead of the product. This choice gives us the capability of using only one 8-bit register for the product instead of $n \times$ 8-bit registers in high-speed architecture. Clearly, this architectural modification results in exploiting only one 8-bit adder as well. By decreasing the number of adder and register pairs from n to only one, the required area and corresponding power consumption are decreased. As mentioned earlier, the multiplication requires n rounds to be completed. This architectural modification causes each round of the multiplication to require n clock cycles in the lightweight architecture. Hence, by taking the very final addition to account, the total number of cycles for this architecture to perform decryption is $(n+1) \times n$.

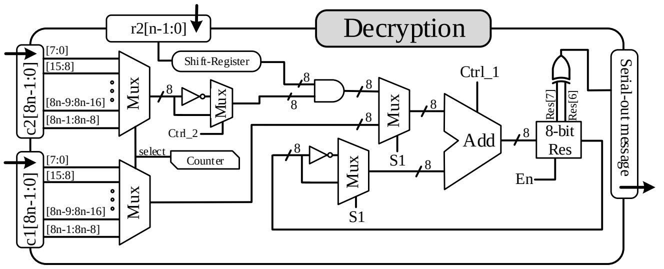

Fig. 5. Ultra-Lightweight Architecture for Decryption Phase.

All of the control signals are very similar to the high-speed architecture with the difference that a counter to keep track of each state is added. To perform anti-circular rotation in shift-and-add algorithm over the ring $\mathcal{R}_q$, two additional control signals *Crtl_1* and *Crtl_2* are added, which decide whether the current operation is an *addition* or a *subtraction*.

C. Side-Channel Analysis (SPA and Timing)

The main target for the proposed architectures is IoT devices that are mostly provided by constrained resources. Therefore, no additional countermeasures are used in the proposed architectures. We aim to provide SPA and timing attack resistant architectures according to principals proposed by [26]. We note that regarding stronger attacks such as differential power analysis (DPA), complex and resource-consuming countermeasures that are proposed in previous work such as [25], [31] can also be applied to our architectures and can be considered for future work.

In the proposed architecture for the InvRBLWE scheme, there are no conditional branches that is taken or not taken according to a secret value. Therefore, the architecture operates independently from input values during each clock cycle, which results in constant number of clock cycles too. More precisely, regarding high-speed architecture, the key generation, encryption and decryption phases require exactly $n+1$, $2n+3$ and $n+1$ clock cycles, respectively. Moreover, using the ultra-lightweight architecture, the key generation, encryption and decryption phases require exactly $(n+1) \times 256$, $(2n+3) \times 256$ and $(n+1) \times 256$ clock cycles, respectively. In addition, the CPD of the proposed architectures is always the same during each phase of execution. Thus, the proposed architecture is secure against timing attacks [27].

To quantitatively evaluate the SPA resistance, we implemented InvRBLWE on a Sakura-X board in order to capture power traces. The power measurements have been performed using a *deep-memory high-performance* USB oscilloscope

TABLE I
COMPARISON OF FPGA IMPLEMENTATION RESULTS

scheme	work	Bit Sec.[†] Qtm/Clsc	Device△	LUT	FF	Slice	DSP/ BRAM	CCs Enc/Dec•	Freq(MHz) Enc/Dec•	Time(µs) Enc/Dec•	AT◇ Enc/Dec•	Improvement∓ Enc/Dec•	
												Time	AT
ECC GF(2^{163})	[6]	-/80	Vrtx-7	9k	2k	2435	0/0	-	264/264	3.01/3.01	7329/7329	68/82	92/95
				23k	6k	5753	0/0	-	214/214	1.50/1.50	8630/8630	37/64	93/97
ECC GF(2^{283})	[6]	-/128	Vrtx-7	20k	3k	5417	0/0	-	226/226	6.01/6.01	32k/32k	67/84	97/97
				45k	11k	11637	0/0	-	185/185	2.99/2.99	34k/34k	33/68	98/98
R-BinLWE	[25]	73/84	Sprt-6	6k	6k	1874	0/0	-/262	-/101	-/2.59	-/4859	-/68	-/87
				57	30	19	0/2	-/65k	-/135	-/486.9	-/9251	-/99	-/93
InvRBLWE $n = 256$	This Work	73/84	Sprt-6	2k	2k	747	0/0	512/256	161/309	3.2/0.82	2390/613		
			Vrtx-6	2k	2k	652	0/0	512/256	456/472	1.1/0.54	717/352		
			Vrtx-7	2k	2k	652	0/0	512/256	540/560	0.95/0.46	619/300		
Ring-LWE	[17]	80/106	Vrtx-6	1k	860	313	1/2	6k/2k	313/313	20.1/9.1	6k/3k	85/83	54/52
	[24]			4k	3k	1137	1/12	6k/4k	262/262	26.2/16.8	29k/19k	92/92	89/92
NTRU	[28]	-/108	Vrtx-E	-	-	6373	0/0	259/-	50/-	5.2/-	33k/-		90/-
Ring-LWE	[17]	120/157	Vrtx-6	1k	950	384	1/3	13k/5k	278/278	47.9/21	18k/8k	95/95	83/81
	[24]			5k	4k	1399	1/14	13k/8k	251/251	54.8/35.4	76k/49k	99/99	99/99
Isogeny⊕ $p = 751$	[29]	124/186	Vrtx-7	44k	51k	16k	376/56	-	-/198	-/9k	-/144m	-/99	-/99
McBits⊕	[30]	128/266	UltS+	112k	-	-	-/375	-	-/225	-/4k	-	-/99	-
InvRBLWE $n = 512$	This Work	140/190	Sprt-6	5k	5k	1464	0/0	1k/512	139/301	7.4/1.62	10k/2372		
			Vrtx-6	5k	5k	1292	0/0	1k/512	443/455	2.32/1.13	2997/1460		
			Vrtx-7	5k	5k	1292	0/0	1k/512	524/541	1.97/0.95	1253/1227		

[†] "Bit Sec. Qtm/Clsc" stands for quantum/classic bits of security. The schemes are sorted in order of their (approximate) quantum security levels according to previous analysis [10]–[12], [15], [22], [28], [29].

△ Vrtx, Sprt and Ults+ stand for Virtex, Spartan and Ultrascale+, respectively.

◇ AT = number of slices × time in microseconds.

⊕ The reported results are for key generation operation.

∓ Improvement = $\frac{x-y}{x} \times 100$, where x and y are related to previous and current work, respectively.

• For all the InvRBLWE scheme, the key generation and decryption operations take the same time; hence one is only reported.

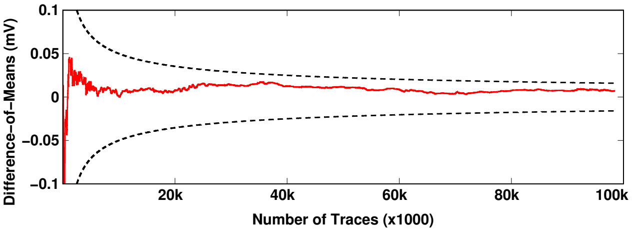

Fig. 6. Quantitative Analysis Regarding SPA Resistance.

(PicoScope 6403D) that can sample at 5 *GS/s* rate. Using the same method as described in [25], we performed difference-of-means test for up to 100,000 traces on $r_2[2]$ (the input key for decryption) while $r_2[1:0]$ is fixed to '10' same as [25]. As shown in Fig. 6, SPA is not possible even after 100,000 traces with the confidence of 99.99% (bounded by the dashed lines).

V. IMPLEMENTATION RESULTS

As discussed in previous section, we propose two architectures for InvRBLWE targeting different devices in edge and IoT nodes. We have implemented the high-speed architecture on both FPGA and AISC platforms due to the fact that edge and powerful devices in IoT can take advantage of having: 1) high-performance crypto-processor, 2) system-on-chip (SoC), or 3) FPGA devices. On the other hand, the ultra-lightweight architecture is a proper match for tiny end-nodes and *resource-constrained* devices that in many cases are powered by energy harvesting systems and have limited power thresholds.

In the followings, first, we provide results over three mostly used FPGA devices by previous work. Second, regarding ASIC implementations of proposed architectures, the 65nm *Taiwan Semiconductor Manufacturing Company* (TSMC) [32] and 45nm *Nangate* digital standard cell libraries [33] are used. We compare our ASIC implementation results against ECC [6] and SIKE [29], which suggests that the proposed implementations are proper match for future (quantum) and even current era in IoT.

A. FPGA Implementation

Table I presents the detailed results of comparison between the proposed implementations and previous work (using $n = 256$ and $n = 512$ that provide 84 and 190 bits of classic security, respectively). The evaluations cover all previous LWE-based FPGA implementations. As the underlying scheme is considered as a post-quantum scheme [11], [12], we have also compared our implementations against the submitted post-quantum schemes in the first round submissions held by NIST [16], which provided hardware implementation results. Additionally, to show the practicality of the proposed architec-

TABLE II
COMPARISON OF ASIC IMPLEMENTATIONS

Scheme	Bits‡ of Security	Tech. (nm)	Freq. (MHz)	Area		Power• (mW)	Time (µs) Enc/Dec	Energy (nj) Enc/Dec
				μm^2	GE			
ECC [6] $GF(2^{163})$	80	65	6.81	29k	20k	1.12	116 / 116	130 / 130
			4.06	40k	28k	1.12	194 / 194	219 / 219
InvRBLWE-HighSpeed (n=256)	84	45	10	28k	32k	0.77	51 / 26	41 / 20
		65	10	46k	32k	1.30	51 / 26	66 / 33
InvRBLWE-Lightweight (n=256)	84	45	33.3	6k	7.6k	0.18	3.8k / 1.9k	713 / 355
		65	33.3	11k	7.9k	0.38	3.8k / 1.9k	1.4k / 722
ECC [6] $GF(2^{283})$	140	65	6.22	49k	34k	1.70	218 / 218	366 / 366
			4.01	72k	50k	1.50	138 / 138	211 / 211
Isogeny [29] $p = 751$	186	65	350	1.7m	1.2m	-	- / 5140	-
InvRBLWE-HighSpeed (n=512)	190	45	10	51k	64k	1.5	102 / 51	171 / 81
		65	10	92k	64k	2.5	102 / 51	225 / 112
InvRBLWE-Lightweight (n=512)	190	45	33	7.9k	9.9k	0.28	15.2k / 7.6k	4k / 2k
		65	33	15k	10.3k	0.70	15.2k / 7.6k	10.5k / 5.3k

‡ The schemes are sorted in order of their classic security levels.
• The power reported in the paper is calculated as "total power = static power + dynamic power".

tures, we compare our implementation results against the best implementations of ECC as a classic cryptosystem. We exploit $Area \times Time$ as the measurement for complexity, which is a common and widely used measurement [6] in previous work.

In [25], three hardware implementations for decryption phase of standard Ring-BinLWE [11] on Xilinx Spartan-6 device for scheme parameters $n = 256$ and $q = 256$ have been provided. Our implementation using the optimized InvRBLWE scheme benefit from avoiding reduction and therefore, have shorter CPD. This makes our implementations with the same set of parameters on the same device to have up to 3 times higher frequency and perform about 3.1 times faster than the *high-performance* implementation in [25]. Thus, InvRBLWE improves decryption time and AT complexity by at least 68% and 87% compared to the fastest implementation in [25], respectively.

Compared to standard Ring-LWE [10] implementations [17], [24], InvRBLWE improves time and AT complexity by at least 83% and 52% on Xilinx Virtex-6 device. Moreover, compared to the best of the available hardware implementations of other post-quantum cryptosystems, our maximum security implementation ($n = 512$ and $q = 256$) achieves higher frequency while consuming less slices of FPGA and improves AT complexity significantly.

Finally, compared to the most recent and the best of the ECC's implementations [6], we achieve higher speed and more than 92% AT improvement. Regarding time and AT comparison, we have compared our most secure implementation (190 bits of security) against the ECC scheme over $GF(2^{283})$ in order to offer a fair result.

B. ASIC Implementation

To show that the proposed hardware architectures are platform-independent, we have implemented them on ASIC using two different 65nm TSMC and 45nm Nangate standard cell libraries. To the best of our knowledge, this is the first ASIC implementation of an LWE-based cryptosystem.

According to the recent NIST's lightweight crypto standardization reports [3], a practical crypto-processor for resource-constrained IoT end-nodes should meet certain power and energy thresholds. Battery-powered devices have limited energy in contrast to energy-harvesting devices that can provide unlimited energy with bounded power.

Table II compares our ASIC implementation results against those for SIKE [29] as a post-quantum key generation scheme based on isogenies over elliptic curve. Moreover, the best ASIC implementation results available for ECC have been compared against ours. Due to the low-complexity operations of the optimized InvRBLWE scheme, our ASIC implementation is at least two times faster than ECC implementations. Additionally, the ultra-lightweight ASIC implementation consumes 62% and 66% less area and power compared to the most lightweight implementation of ECC, respectively.

As seen in the table, our high-speed architecture provides fast scheme operations compared to all of previous work, while consuming lower energy. This makes the high-speed implementation of InvRBLWE a proper choice for IoT devices powered by batteries. On the other hand, the ultra-lightweight architecture can be implemented by only 7.5k gates and also consumes only 0.18 mW. Such low power consumption makes this architecture to be the first public key implementation that can be supplied by ultra low-power energy harvesters such as Vibration Piezo or EM [3].

We note that the ASIC implementation results for both architectures indicate that InvRBLWE has the potential to be considered as an alternative for classic cryptosystems.

VI. CONCLUSION

Security of IoT is becoming a concern considering the exponential growth in IoT nodes and applications. Moreover, post-quantum security is also an issue for the future of internet and IoT due to the venerability of the classic cryptosystems such as ECC and RSA against quantum attacks [7]. Thus, many of the organizations are already looking for reliable and practical alternatives for classic cryptosystems [16], [34]. Among current post-quantum crypto-schemes, lattice-based cryptography has gained high attention from researchers due to its relatively faster operations compared to other post-quantum cryptosystems such as code-based [13] or isogeny [14].

In this paper, we propose an optimized version of Ring-BinLWE [11], namely InvRBLWE, which is highly efficient for hardware implementation. Moreover, we propose two architectures for InvRBLWE scheme targeting different IoT devices with alternative capabilities varying from edge and powerful devices to resource-constrained and tiny end-nodes. Our FPGA implementations improve time and AT compared to the best previous RingLWE implementations [17], [24], [25] by at least 68% and 52%, respectively. These implementations also improve AT by at least 92% compared to the best of ECC implementations. We are the first to propose a practical ASIC implementation of LWE-based cryptosystems for IoT devices. The proposed high-speed ASIC implementation is at least two times faster than the best of the ECC implementations while consuming less energy. This makes the high-speed architecture suitable for battery-based solutions. Moreover, our lightweight ASIC implementation has power consumption as low as 0.18mW, which can be supplied by low-power energy harvesting devices such as Vibration Piezo or EM [3]. In other words, the lightweight architecture is the first public key ASIC implementation that satisfies NIST report criteria [3] and can be practically exploited in IoT end-nodes.

As future work, the proposed hardware implementations can be extensively analyzed against side-channel analysis (SCA). Our current implementations are resistant regarding SPA and timing attacks. However, there exist other powerful attacks such as differential power analysis (DPA) [26], simple and differential fault attacks (SFA and DFA) [31].

Acknowledgment

We would like to thank the reviewers for their insightful comments. This research was partially supported by Sharif University of Technology under the grant number G960803.

References

[1] Z. Ling, J. Luo, Y. Xu, C. Gao, K. Wu, and X. Fu, "Security vulnerabilities of internet of things: A case study of the smart plug system," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1899–1909, Dec 2017.

[2] 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 of Things Journal*, pp. 1–1, 2018.

[3] C. Patrick and P. Schaumont, "The role of energy in the lightweight cryptographic profile," in *NIST Lightweight Cryptography Workshop*, 2016.

[4] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[5] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[6] R. Salarifard, S. Bayat-Sarmadi, and H. Mosanaei-Boorani, "A low-latency and low-complexity point-multiplication in ecc," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. PP, no. 99, pp. 1–9, 2018.

[7] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*. Ieee, 1994, pp. 124–134.

[8] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM, 2009, pp. 333–342.

[9] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM (JACM)*, vol. 56, no. 6, p. 34, 2009.

[10] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2010, pp. 1–23.

[11] J. Buchmann, F. Göpfert, T. Güneysu, T. Oder, and T. Pöppelmann, "High-performance and lightweight lattice-based public-key encryption," in *Proceedings of the 2nd International Workshop on IoT Privacy, Trust, and Security*. USA: ACM, 2016, pp. 2–9.

[12] J. Buchmann, F. Göpfert, R. Player, and T. Wunderer, "On the hardness of lwe with binary error: revisiting the hybrid lattice-reduction and meet-in-the-middle attack," in *International Conference on Cryptology in Africa*. Springer, 2016, pp. 24–43.

[13] "Classic McEliece." [Online]. Available: https://classic.mceliece.org/.

[14] D. Jao and L. De Feo, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," in *International Workshop on Post-Quantum Cryptography*. Springer, 2011, pp. 19–34.

[15] B. Koziel, R. Azarderakhsh, and M. Mozaffari-Kermani, "A high-performance and scalable hardware architecture for isogeny-based cryptography," *IEEE Trans. Computers*, 2018.

[16] "National institute of standards and technology." [Online]. Available: https://www.nist.gov/.

[17] S. S. Roy, F. Vercauteren, N. Mentens, D. D. Chen, and I. Verbauwhede, "Compact ring-lwe cryptoprocessor," in *Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2014, pp. 371–391.

[18] "The cryptographic suite for algebraic lattices (CRYSTALS)." [Online]. Available: https://pq-crystals.org/.

[19] "NTRU Prime family: Streamlined NTRU Prime and NTRU LPRime." [Online]. Available: https://ntruprime.cr.yp.to/index.html.

[20] D. Micciancio and C. Peikert, "Hardness of sis and lwe with small parameters," in *Advances in Cryptology–CRYPTO 2013*. Springer, 2013, pp. 21–39.

[21] R. De Clercq, S. S. Roy, F. Vercauteren, and I. Verbauwhede, "Efficient software implementation of ring-lwe encryption," in *Design, Automation & Test in Europe Conference (DATE)*. IEEE, 2015, pp. 339–344.

[22] F. Göpfert, C. van Vredendaal, and T. Wunderer, "A hybrid lattice basis reduction and quantum search attack on lwe," in *International Workshop on Post-Quantum Cryptography*. Springer, 2017, pp. 184–202.

[23] N. Göttert, T. Feller, M. Schneider, J. Buchmann, and S. Huss, "On the design of hardware building blocks for modern lattice-based encryption schemes," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2012, pp. 512–529.

[24] T. Pöppelmann and T. Güneysu, "Towards practical lattice-based public-key encryption on reconfigurable hardware," in *International Conference on Selected Areas in Cryptography*. Springer, 2013, pp. 68–85.

[25] A. Aysu, M. Orshansky, and M. Tiwari, "Binary ring-lwe hardware with power side-channel countermeasures," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2018, pp. 1253–1258.

[26] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, "Introduction to differential power analysis," *Journal of Cryptographic Engineering*, vol. 1, no. 1, pp. 5–27, 2011.

[27] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Annual International Cryptology Conference*. Springer, 1996, pp. 104–113.

[28] A. A. Kamal and A. M. Youssef, "An fpga implementation of the ntru-encrypt cryptosystem," in *Microelectronics (ICM), 2009 International Conference on*. IEEE, 2009, pp. 209–212.

[29] D. Jao, R. Azardrakhsh, M. Campagna, C. Costello, L. D. Feo, B. Hess, A. Jalali, B. Kozeil, B. LaMacchia, P. Longa, M. Naehrig, J. Renes, V. Soukharev, and D. Urbanik, "Supersingular isogeny key encapsulation (SIKE)."

[30] W. Wang, J. Szefer, and R. Niederhagen, "Fpga-based key generator for the niederreiter cryptosystem using binary goppa codes," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2017, pp. 253–274.

[31] T. Schneider, A. Moradi, and T. Güneysu, "Parti–towards combined hardware countermeasures against side-channel and fault-injection attacks," in *Annual Cryptology Conference*. Springer, 2016, pp. 302–332.

[32] "Semiconductor manufacturing company (TSMC)." [Online]. Available: http://www.tsmc.com/.

[33] "NanGate standard cell library." [Online]. Available: http://www.si2.org/.

[34] L. Chen, L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, *Post-quantum cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2016.