LAC: Practical Ring-LWE Based Public-Key Encryption with Byte-Level Modulus ⋆

Xianhui Lu[1,2,3], Yamin Liu[1,2,3], Zhenfei Zhang[4], Dingding Jia[1,2,3], Haiyang Xue[1,2,3], Jingnan He[1,2,3] and Bao Li[1,2,3]

1. Data Assurance and Communication Security Research Center, Chinese Academy of Sciences (CAS).
2. State Key Laboratory of Information Security, Institute of Information Engineering, CAS.
3. School of Cyber Security, University of Chinese Academy of Sciences.
4. Algorand.
luxianhui@outlook.com, liuyamin@iie.ac.cn, zhenfei@algorand.com

Abstract. Lattice based cryptography is one of the leading candidates of the post quantum cryptography. A major obstacle of deployment, though, is that its payload is relatively larger than the classical solutions, such as elliptic curve Diffie-Hellman. In this paper, we investigate the approach of reducing the key size and ciphertext size by decreasing the size of the modulus, and propose the first instantiation to the family of ring learning with error based solutions where the modulus is at a byte level. The main technical contributions of this paper are around the implementation side of the algorithms. With the use of large-block error correction code, we are able to propose parameter sets with small moduli while achieving a negligible decryption error rate. We investigate best known attacks, and give a concrete security estimation of the proposed parameter sets. Since our parameter sets are no longer compatible with number theoretic transform (NTT), we also present optimizations for ring multiplications. As a result, our scheme is more compact and nearly as efficient as popular solutions in this domain, such as NewHope and Kyber.

Keywords: lattice based cryptography, learning with errors, error correction, NIST post-quantum cryptography standardization.

1 Introduction

Due to the rapid advances of quantum computing, the construction of cryptographic schemes secure against quantum attacks (a.k.a post-quantum cryptography) becomes an important mission in the field of cryptology. Lattice based cryptography is one of the most promising and mature candidates for the post-quantum migration plan of the National Institute of Standards and Technology

⋆ LAC is one of 69 submissions that are under Round 1 evaluation of NIST-PQC [1]. The work was partially done when Zhenfei Zhang was at OnBoard Security Inc.

(NIST) [25,1]. However, a major obstacle of deploying lattice based solutions, other than to understand the concrete security of the scheme, is that the payload sizes (for example, the public key and ciphertext) are much larger than a classical solution. For instance, in a TLS handshake, it is desirable to have the public key and ciphertext size to be less than 1 KB so that the whole hello message fits in a maximum traffic unit.

The Learning With Errors (LWE) problem was initially proposed by Regev [57], and became extremely versatile in constructing public key encryption schemes [56,52,44,49,15], identity based encryption schemes [37,21,2,3] and fully homomorphic encryption schemes [19,18,38]. Despite of all those ground breaking applications, the main drawback remains that they have key size at least quadratic in the main security parameter. Inspired by the NTRU cryptosystem [40] and the ring-based short integer solution problem [48,45], Lyubashevsky, Peikert and Regev [46,47,55] resolved this problem by introducing an algebraic variant of LWE, namely, Ring-LWE; and showed that its security can be reduced to worst-case problems on ideal lattices. It is worth noting that in a concurrent and independent work, Stehlé *et al.* [64] also proposed a special case of Ring-LWE over power-of-two cyclotomic polynomials; in [55], it was shown that Ring-LWE problem is hard for any ring with appropriate error distribution.

For almost all LWE based constructions, there exists an instantiation with Ring-LWE where the size of the public key and the ciphertext can be reduced by a factor of n, where n is the dimension of the polynomial ring. Depending on the choice of the ring, one may also carry out the ring multiplications in $O(n \log q)$ by using the fast Fourier transform (FFT) or number theoretic transform (NTT), where q is the modulus. Due to its great security, utility and efficiency, Ring-LWE and its variants become the most popular building-blocks in the design of practical cryptosystems [16,10,9,17,42,61,11].

To date, the (Ring-)LWE based public key encryption schemes have arrived at a mature state that is almost ready for deployment, except for the aforementioned size problem. For the public key encryption schemes, they all follow a similar framework by Regev [57] and Lyubashevsky *et al.* [46]. For the key exchange protocols, one may use the reconciliation method, first put forth by Ding [29], and then refined by Peikert [53], to improve efficiency.

Subsequent works, such as [16,10,17,61,11], have contributed to a large portion of the NIST post-quantum cryptography standardization process (NIST-PQC) [1]. As one has seen from those work, further improvement of the bandwidth efficiency has become one of the main missions in the design of practical lattice based cryptographic schemes.

1.1 Our Contributions

Motivation. Before presenting our contributions, let us briefly present our motivation. For the sake of simplicity and efficiency, we will use the ring $\mathcal{R} = \mathbb{Z}_q[x]/(x^n + 1)$ with a power-of-two n, a favorable choice by many Ring-LWE based schemes [30,10], to illustrate our idea, although we must remark that it is shown by Peikert *et al.* in [55] that Ring-LWE is hard for any ring of integers.

Intuitively, the hardness of Ring-LWE problem is mainly determined by the error rate α (the ratio of the noise magnitude to the modulus q) and the dimension n. According to the concrete hardness analysis[1] in [8,10,7], suitable choices of the dimension n are $2^9 = 512$ and $2^{10} = 1024$. For these choices, $q = 12289$ is the smallest prime for which $q \equiv 1 \bmod 2n$. In other words, to enable the super efficient NTT multiplications, we have a constraint that q is at least 12289. As a result, $q = 12289$ is now the most widely used modulus for the Ring-LWE based schemes[2].

On the other hand, this constraint is a bit artificial, in that it is purely decided by NTT, and not regulated by any security requirement. To be more specific, the security level grows with the error rate, which is the ratio between the error and the modulus, rather than the modulus itself. Therefore, to achieve a great compactness with an acceptable level of security, it makes sense to choose the modulus as small as possible, while keeping the ratio somewhat a constant.

In this paper, we investigate the approach of reducing the key size and ciphertext size of Ring-LWE based schemes by decreasing the size of the modulus. We consider "byte" level modulus. Byte is the smallest data type that modern processors handle. It seems to be a sweet spot to balance performance, size and security. We also remark that for moduli that are significantly smaller than 256, the performance gain will be minimal (since processors will treat the data type as a byte anyway) while it becomes infeasible to find error distributions that can maintain a same error/modulus ratio.

Parameter Derivation. There have been a sequence of work on the theoretical worst-case hardness of Ring-LWE problems [64,46,55,59]. However, they give no guidance on the choice of concrete parameters. Parameter derivation is an active research topic for lattice based cryptography, for both cryptography and cryptanalysis [26,54,5]. Arguably, most lattice based submissions to NIST-PQC follow a similar design [46,44,29,53], and a major differentiator among the schemes is the choices of parameters.

As mentioned earlier, we consider the family of "byte" level modulus that breaks the constraint of NTT modulus. Specifically, we consider three types of byte-level moduli, namely "power-of-two modulus", "max-split modulus" and "min-split modulus". We then select proper secret and error distribution to match the proposed modulus. Since the concrete security partially relies on the error rates, to be able to sample errors efficiently becomes crucial to the overall design. For provable security one requires discrete Gaussian samples; however, in practice it is sufficient to sample from distributions that are close enough to a Gaussian. We observe that centered binomial distribution with the standard deviation of $\sigma = 1/\sqrt{2}$ is a sweet spot for security, correctness and efficiency.

[1] As opposite to the provable security, this is a method to obtain the bit-complexity by looking at the cost of best known attacks, such as BKZ with quantum sieving.

[2] Note that, Kyber [17], a Module-LWE based scheme, uses a smaller polynomial ring of degree $n = 256$. NTT over this ring is possible with a smaller modulus $q = 7681$.

To show the concrete security of our scheme, similar to other works in this field, we perform a concrete analysis of best known attacks, using both the popular and generic BKZ analysis with (quantum) sieving [8,10,7], as well as dedicated attacks, such as the subfield attacks and hamming weight attacks [51].

Error Corrections. In most lattice based schemes, dated back from one of the first lattice based encryption schemes, NTRU [40], there exists a (tunable) decryption error probability. One may choose a zero decryption error probability, at the cost of a larger modulus (and hence larger keys and ciphertexts); or a negligible one, with a moderate size modulus. See, for example [34], for a comparison of different error correction codes for lattice based cryptography. Our byte level modulus incurs a very high decryption error rate by design; and simple error correction techniques, such as D2 or D4 codes [10], do not work well in our use case.

To cope with this noise growth, as well as the byte level modulus, we encode the plaintext message with an error correction code that supports very large block size. Generally speaking, with the great power comes great cost: error correction code for large block sizes brings severe efficiency penalty. We propose to use binary BCH error correction code, which is particularly efficient, in both encoding and decoding, when the code size is smaller than 1023 and the number of error bits does not exceed 50. With BCH code we are able to decrease the decryption error rate to a desirable level.

Note that, the choice of error correction code will not affect the security of the scheme (see Section 4.4 for more details). Our scheme in principle supports any error correction code with required error correction ability.

Implementation. Recall that we have switched to a byte level modulus, we can no longer resort to NTT for efficient ring multiplications. Popular alternatives are Karatsuba/Toom-Cook algorithms, such as [13,27] and index based multiplications algorithms, such as [11]. We adopt the index based solutions since they work particularly well with ternary secret and noise that can be sampled efficiently from binomial distribution with $\sigma = 1/\sqrt{2}$. In addition, with byte level moduli and bit-level noises, mod operation may be called less often: it is not essential to call it after every arithmetic operations; it can be called only when necessary.

Byte-level modulus also brings more parallelization. We implement polynomial multiplication with the vector instructions, for example, AVX2 over the Intel64 platform. As a result, our optimized implementation over this platform is nearly as efficient as popular schemes with NTT [10,17], despite of the heavy penalty from error correction codes.

Comparison. To highlight the compactness and efficiency of our scheme, we briefly compare the performance of our scheme with NewHope [9] and Kyber

[17]. We compare the chosen ciphertext secure version of the schemes. Note that, there are several versions of parameter sets for both NewHope and Kyber. Here, we use their recommended parameter submitted to NIST-PQC [1]. Cycle counts of NewHope and Kyber were obtained on an Intel Core i7-4770K (Haswell) with Turbo Boost and Hyperthreading disabled. Cycle counts of LAC was obtained on an Intel Core i7-4770S (Haswell) with Turbo Boost and Hyperthreading disabled. In terms of security, NewHope and Kyber [9,17] uses BKZ with (quantum) sieving to estimate their security. We follow a same method, and our estimation is independently confirmed in [5].

Scheme	**Size (in Bytes)**			**AVX2 Cycles**			**Security**
	sk	**pk**	**ct**	**gen**	**enc**	**dec**	
NewHope512	1888	928	1120	68,080	109,836	114,176	101
Kyber512	1632	736	800	55,160	75,680	74,428	102
LAC-128v2	1056	544	736	53,062	72,382	92,203	133
Kyber768	2400	1088	1152	85,472	112,660	108,904	161
LAC-192v2	2080	1056	1248	108,900	171,042	244,043	259
NewHope1024	3680	1824	2208	129,670	210,092	220,864	233
Kyber1024	3168	1440	1504	121,056	157,964	154,952	218
LAC-256v2	2080	1056	1472	97,464	174,512	323,207	290

sk secret key **pk** public key
ct ciphertext **gen** key generation
enc encryption or encapsulation **dec** decryption or decapsulation

Table 1. Comparison of NewHope, Kyber and LAC

To the best of our knowledge, our new scheme is the first Ring-LWE based public key encryption scheme with byte-level modulus. A previous version of our scheme named "LAC" was submitted to NIST-PQC, and is one of 69 submissions that are under Round 1 evaluation [1]. In this paper, we revise the proposed parameter sets. Looking ahead, our proposed parameter sets, namely LAC-128v2, LAC-192v2 and LAC-256v2 are identical to previous version, except for a different message length and consequently a different error correction parameter. This modification allows for even lower decryption error rates, and better addresses the high Hamming weight attacks [51].

1.2 Outline

In section 2 we describe basic definitions and notations. In section 3 we describe our Ring-LWE based public key encryption scheme. In section 4 we describe the selection of parameters. In section 5 we give the security evaluation of our new scheme. In section 6 describe the optimized implementation of our scheme. Finally, we give the conclusion in section 7.

2 Preliminaries

In this section we first define several mathematical notations, the definitions of Ring-LWE and public key encryption schemes.

2.1 Basic Notations

Vectors are denoted by bold lower-case characters, such as $\boldsymbol{a}$. $\boldsymbol{a}^t$ denotes the transposition of $\boldsymbol{a}$. Matrices are denoted by upper-case characters, such as $\boldsymbol{A}$. $\boldsymbol{A}^t$ denotes the transposition of $\boldsymbol{A}$. For an m-dimensional vector $\boldsymbol{a} = (a_1, a_2, ..., a_m)$, the l_2-norm, also known as the Euclidean norm, is defined as $\|\boldsymbol{a}\| := \sqrt{\sum_{i=1}^{m} a_i^2}$. The length of a matrix is the norm of its longest column vector, e.g., $\|\boldsymbol{A}\| := \max \|\boldsymbol{a}_i\|$. For an m-dimensional vector $\mathbf{a} = (a_0, \cdots, a_{m-1})$ and a non-negative integer $l \leq m$, define $(\mathbf{a})_l := (a_0, \cdots, a_{l-1})$.

For a set S, $x \xleftarrow{\$} S$ denotes that an element x is chosen from S uniformly at random. For a distribution D, $x \xleftarrow{\$} D$ denotes that a random variable x is sampled according to D. For a randomized algorithm A, $y \xleftarrow{\$} \mathsf{A}(x)$ denotes that y is assigned randomly from the set of output of A with input x; if the algorithm A is deterministic, we simplify it as $y \leftarrow \mathsf{A}(x)$.

For an integer $q \geq 1$, let $\mathbb{Z}_q$ be the residue class ring modulo q, define the ring of integer polynomials modulo $x^n + 1$ as $\mathcal{R} := \mathbb{Z}[x]/(x^n + 1)$ for an integer $n \geq 1$, and the ring $\mathcal{R}_q := \mathbb{Z}_q[x]/(x^n + 1)$ denotes the polynomial ring modulo $x^n + 1$ where the coefficients are from $\mathbb{Z}_q$.

2.2 Distributions and Random Sampling

The Uniform Distribution. The uniform distribution over a set X is defined as $U(X)$. For example, the uniform distribution over $\mathcal{R}_q$ is $U(\mathcal{R}_q)$.

The Centered Binomial Distribution. The idea to simulate a Gaussian distribution with binomial distribution was firstly introduced in [10], in order to mitigate the heavy cost of Gaussian sampling. In the design of LAC we also use centered binomial distribution with parameter 1 and $\frac{1}{2}$ (denoted by Ψ_1 and $\Psi_{\frac{1}{2}}$, respectively) as follows:

Definition 1 (Ψ_1). *Sample $(a, b) \xleftarrow{\$} \{0, 1\}^2$, and output $a - b$. It picks 0 with probability $\frac{1}{2}$, and ± 1 with probability $\frac{1}{4}$ according to the distribution Ψ_1. The mean value of Ψ_1 is 0 and the variance is $\frac{1}{2}$.*

Definition 2 ($\Psi_{\frac{1}{2}}$). *Sample $(a, b) \xleftarrow{\$} \Psi_1$, and output $a * b$. It picks 0 with probability $\frac{3}{4}$, and ± 1 with probability $\frac{1}{8}$ according to the distribution $\Psi_{\frac{1}{2}}$. The mean value of $\Psi_{\frac{1}{2}}$ is 0 and the variance is $\frac{1}{4}$.*

Random Sampling. Denote by Samp an abstract algorithm that samples a random variable according to a distribution with a given seed:

$$x \leftarrow \mathsf{Samp}(D; \mathsf{seed}),$$

where D is a distribution, and seed is the random seed used to sample x. For an empty $\mathsf{seed} = \perp$, the process is randomized, and equivalent to $x \xleftarrow{\$} D$. When a seed is present, the sampling of x will be deterministic.

We extend the definition to a multiple dimension setting. We use

$$(x_1, x_2, \cdots, x_t) \leftarrow \mathsf{Samp}(D_1, D_2, \cdots, D_t; \mathsf{seed})$$

to denote the process of sampling random variables x_i-s from distributions D_i-s for $1 \leq i \leq t$.

2.3 Learning with Errors (over Rings)

We refer the readers to [57,58,64,46,55] for a concrete background of the definitions and reductions.

Definition 3 (Search LWE). *Let n, m, q be positive integers, and $\chi_{\boldsymbol{s}}, \chi_{\boldsymbol{e}}$ be (bounded) distributions over $\mathbb{Z}$. Given $(\boldsymbol{A}, \boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e})$, recover the secret $\boldsymbol{s}$, where $\boldsymbol{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, the secret $\boldsymbol{s} \xleftarrow{\$} \chi_{\boldsymbol{s}}^n$ and the error $\boldsymbol{e} \xleftarrow{\$} \chi_{\boldsymbol{e}}^m$.*

Definition 4 (Decisional LWE). *Let n, m, q be positive integers, and $\chi_{\boldsymbol{s}}, \chi_{\boldsymbol{e}}$ be (bounded) distributions over $\mathbb{Z}$. Distinguish the two distributions of $(\boldsymbol{A}, \boldsymbol{b})$ and $(\boldsymbol{A}, \boldsymbol{u})$, where $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}$ for $\boldsymbol{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, $\boldsymbol{s} \xleftarrow{\$} \chi_{\boldsymbol{s}}^n$, $\boldsymbol{e} \xleftarrow{\$} \chi_{\boldsymbol{e}}^m$, $\boldsymbol{u} \xleftarrow{\$} \mathbb{Z}_q^m$.*

Definition 5 (Search Ring-LWE). *Let n, q be positive integers, and $\chi_{\boldsymbol{s}}, \chi_{\boldsymbol{e}}$ be (bounded) distributions over $\mathcal{R}$. Given $(\boldsymbol{a}, \boldsymbol{b} = \boldsymbol{a}\boldsymbol{s} + \boldsymbol{e})$, recover the secret $\boldsymbol{s}$, where $\boldsymbol{a} \xleftarrow{\$} \mathcal{R}_q$, the secret $\boldsymbol{s} \xleftarrow{\$} \chi_{\boldsymbol{s}}$ and the error $\boldsymbol{e} \xleftarrow{\$} \chi_{\boldsymbol{e}}$.*

Definition 6 (Decisional Ring-LWE). *Let n, q be positive integers, and $\chi_{\boldsymbol{s}}, \chi_{\boldsymbol{e}}$ be (bounded) distributions over $\mathcal{R}$. Distinguish two distributions of $(\boldsymbol{a}, \boldsymbol{b})$ and $(\boldsymbol{a}, \boldsymbol{u})$, where $\boldsymbol{b} = \boldsymbol{a}\boldsymbol{s} + \boldsymbol{e}$ for $\boldsymbol{a} \xleftarrow{\$} \mathcal{R}_q$, $\boldsymbol{s} \xleftarrow{\$} \chi_{\boldsymbol{s}}$, $\boldsymbol{e} \xleftarrow{\$} \chi_{\boldsymbol{e}}$, $\boldsymbol{u} \xleftarrow{\$} \mathcal{R}_q$.*

2.4 Public Key Encryption

A *public key encryption scheme* PKE=(Gen,Enc,Dec) with message space $\mathcal{M}$ consists of three polynomial-time algorithms.

- **KG**(l): A probabilistic polynomial-time key generation algorithm takes as input the security parameter l and outputs a public key pk and a private key sk. We write $(pk, sk) \leftarrow \mathrm{KG}(l)$.
- **Enc**(pk, m): A probabilistic polynomial-time encryption algorithm takes as inputs a public key pk, a plaintext m and outputs a ciphertext c. We write $c \leftarrow \mathrm{E}_{pk}(m)$.

– **Dec**(sk, c): A decryption algorithm takes as inputs a ciphertext c and a private key sk, and outputs a plaintext m. We write $m \leftarrow \mathrm{D}_{sk}(c)$.

A public key encryption scheme is IND-CCA2 (indistinguishable against adaptive chosen ciphertexts attacks) secure if the advantage of any adversary $\mathcal{A}$ defined in the following is negligible in the security parameter l:

$$\mathrm{Adv}^{\mathrm{cca}}_{\mathcal{A}}(l) = \left| \Pr \left[b' = b : \begin{array}{l} (pk, sk) \leftarrow \mathrm{Gen}(l), \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathrm{D}_{sk}(\cdot)}(pk), \\ b \xleftarrow{R} \{0,1\}, \\ c^* \leftarrow \mathrm{E}_{pk}(m_b), \\ b' \leftarrow \mathcal{A}^{\mathrm{D}_{sk}(\cdot)}(pk, c^*). \end{array} \right] - \frac{1}{2} \right|,$$

where $\mathcal{A}$ is restricted not to query $\mathrm{D}_{sk}(\cdot)$ with c^*.

3 The LAC scheme

In this section we describe our Ring-LWE based public key encryption scheme "LAC". LAC is a concrete instantiation of the Ring-LWE based scheme proposed in [46], and the main deviation at an algorithmic level is that the plaintext message is encoded with a large-block error correction code.

3.1 The scheme

The algorithm LAC.KG randomly generates a pair of public key and secret key (pk, sk).

Algorithm 1 LAC.KG()

Ensure: A pair of public key and secret key (pk, sk).

1: $\mathsf{seed}_{\boldsymbol{a}} \xleftarrow{\$} \mathcal{S}$
2: $\boldsymbol{a} \leftarrow \mathsf{Samp}(U(\mathcal{R}_q); \mathsf{seed}_{\boldsymbol{a}}) \in \mathcal{R}_q$
3: $\boldsymbol{s} \xleftarrow{\$} \Psi_\sigma^n$
4: $\boldsymbol{e} \xleftarrow{\$} \Psi_\sigma^n$
5: $\boldsymbol{b} \leftarrow \boldsymbol{a}\boldsymbol{s} + \boldsymbol{e} \in \mathcal{R}_q$
6: **return** $(pk := (\mathsf{seed}_{\boldsymbol{a}}, \boldsymbol{b}), sk := \boldsymbol{s})$

The algorithm LAC.Enc on input pk and a message $\boldsymbol{m}$, encrypts $\boldsymbol{m}$ with the randomness seed. The subroutine ECCEnc converts the message $\boldsymbol{m}$ into a codeword.

Algorithm 2 LAC.Enc($pk = (\mathsf{seed}_{\boldsymbol{a}}, \boldsymbol{b}), \boldsymbol{m} \in \mathcal{M}; \mathsf{seed} \in \mathcal{S}$)

Ensure: A ciphertext $\boldsymbol{c}$.

$\boldsymbol{a} \leftarrow \mathsf{Samp}(U(\mathcal{R}_q); \mathsf{seed}_{\boldsymbol{a}}) \in \mathcal{R}_q$
$\widehat{\boldsymbol{m}} \leftarrow \mathsf{ECCEnc}(\boldsymbol{m}) \in \{0,1\}^{l_v}$
$(\boldsymbol{r}, \boldsymbol{e}_1, \boldsymbol{e}_2) \leftarrow \mathsf{Samp}(\Psi_\sigma^n, \Psi_\sigma^n, \Psi_\sigma^{l_v}; \mathsf{seed})$
$\boldsymbol{c}_1 \leftarrow \boldsymbol{a}\boldsymbol{r} + \boldsymbol{e}_1 \in \mathcal{R}_q$
$\boldsymbol{c}_2 \leftarrow (\boldsymbol{b}\boldsymbol{r})_{l_v} + \boldsymbol{e}_2 + \lfloor \frac{q}{2} \rceil \cdot \widehat{\boldsymbol{m}} \in \mathbb{Z}_q^{l_v}$
return $\boldsymbol{c} := (\boldsymbol{c}_1, \boldsymbol{c}_2) \in \mathcal{R}_q \times \mathbb{Z}_q^{l_v}$

The algorithm LAC.Dec on input sk and a ciphertext $\boldsymbol{c}$, recovers the corresponding message $\boldsymbol{m}$. The subroutine ECCDec on input an encoding $\widehat{\boldsymbol{m}}$, decoding the codeword in it. Usually, a message $\boldsymbol{m} \in \mathcal{M}$ is recovered.

Algorithm 3 LAC.Dec($sk = \boldsymbol{s}, \boldsymbol{c} = (\boldsymbol{c}_1, \boldsymbol{c}_2)$)

Ensure: A plaintext $\boldsymbol{m}$.

$\boldsymbol{u} \leftarrow \boldsymbol{c}_1 \boldsymbol{s} \in \mathcal{R}_q$
$\widetilde{\boldsymbol{m}} \leftarrow \boldsymbol{c}_2 - (\boldsymbol{u})_{l_v} \in \mathbb{Z}_q^{l_v}$
for $i = 0$ to $l_v - 1$ **do**
  if $\lfloor \frac{q}{4} \rceil \leq \widetilde{\boldsymbol{m}}_i < \lfloor \frac{3q}{4} \rceil$ **then**
    $\widehat{\boldsymbol{m}}_i \leftarrow 1$
  else
    $\widehat{\boldsymbol{m}}_i \leftarrow 0$
  end if
end for
$\boldsymbol{m} \leftarrow \mathsf{ECCDec}(\widehat{\boldsymbol{m}})$
return $\boldsymbol{m}$

3.2 Security proof

Here we mainly consider the formal security, i.e. the security reduction from LAC to the underlying Ring-LWE assumption and the reduction from Ring-LWE problem to basic problems over ideal lattices. Security with regard to concrete parameters shall be presented in Section 5.

Following the result of [44], the chosen plaintext security of LAC can be easily reduced to the Ring-LWE assumption. Then, with Fujisaki-Okamoto transformation, we obtain the chosen ciphertext security version of LAC in both classical random oracle model [35,36] and quantum random oracle model [41]. It is easy to verify that the embedded error correction code will not affect the security reduction and these security proofs can be directly extended to the case of LAC. Therefore, we omit the details for both reductions.

With regard to the second reduction, there has been a few variants of Ring-LWE problems, such as primal-Ring-LWE [31,32,23,24,22,54], dual-Ring-LWE [46,47,55], and Poly-LWE [29,53,16,10,9,17,42,61]. Recently, Rosca *et al.* [59] showed reductions among all of above variants with limited parameter losses.

Thus, it is sufficient to build our scheme from Ring-LWE while enjoying a reduction to the hard problems over ideal lattices, asymptotically speaking.

Remark 1. In practice, it is desirable to sample secrets $\boldsymbol{s}$ and noise $\boldsymbol{e}$ from small distributions, such as a binary or a ternary distribution. Although in [50] is has been shown that plain LWE problem with such choices can be provably secure when the number of samples are limited, their result does not apply to Ring-LWE. It is still an open problem to prove the hardness of Ring-LWE for small parameters and practical number of samples.

4 Parameter Selection

Almost all lattice based key exchanges and public key encryptions, except for NTRU based ones, follow a similar framework from [29,53,16,9]. We have a set of theoretical results on the choice of rings, moduli, errors, etc [55,54,59] that ensure the framework stems from a provable secure design. However, those theoretical results do not give any guidance on selecting concrete parameters. Choosing parameters for (Ring-)LWE based schemes becomes one of a main research direction in subsequent works [16,9,54,17,42,61], and a main differentiator in most NIST-PQC submissions [1]. In this section, we present our choice of parameters, and give our design rational over common choices.

4.1 Modulus

Our first and foremost priority is to reduce the modulus. As mentioned earlier, the payload sizes are governed mainly by the dimension and the modulus. The choice of power-of-2 cyclotomic polynomial does not allow much freedom in the choice of n. Hence we focus on a small modulus to reduce the payload size. Note that the modulus cannot be too small; it needs to be large enough to tolerant the errors during decryption which will be scaled by a factor of $\sqrt{2n}$. A common choice was $q = 12289$. We take a more aggressive approach by using "byte level modulus".

A byte is the basic operating unit for most processors. Such a choice makes the public keys and ciphertexts compact, and is also optimal for implementations. The downside is that decryption errors increase when modulus is smaller. We will give more details in Section 4.3.

Depending on the structure of the polynomial ring, we consider three types of byte-level modulus.

- **Power of Two Modulus:** From the view of implementation, the most suitable byte-level modulus is $q = 256$, for which the modulus operation can be efficiently realized by ignoring the carrier data. However, since $q = 256$ is not a prime, $\mathbb{Z}_{256}[x]/(x^n + 1)$ does not yield a field for our choice of n. For conservative purpose we do not use this ring to avoid any potential weakness of the underlying structure.

– **Max-Split Modulus:** The reason to choose $q \equiv 1 \bmod 2n$ is that $x^n + 1 \in \mathbb{Z}_q[x]$ can be completely factorized. For byte-level modulus, this is no longer the case. However, we notice that when $q = 257$, $x^n + 1 \in \mathbb{Z}_{257}[x]$ has maximum number of factors:

$$x^{512} + 1 = \prod_{i=1}^{128}(x^4 + \tau_i), \qquad x^{1024} + 1 = \prod_{i=1}^{128}(x^8 + \tau_i),$$

where $\tau_i \in \mathbb{Z}_q$. We call this type of modulus "Max-Split Modulus", for which $x^n + 1$ can be maximally factorized into polynomials with very small degrees.

– **Min-Split Modulus:** Unlike $q = 257$, for some other modulus, $x^n + 1 \in \mathbb{Z}_q[x]$ may have minimum number of factors. Concretely, we notice that for $q = 251$, which is the largest prime smaller than 2^8, $x^n + 1 \in \mathbb{Z}_{257}[x]$ can be minimally factorized as:

$$x^n + 1 = (x^{n/2} + 91x^{n/4} + 250)(x^{n/2} + 160x^{n/4} + 250).$$

We call this type of modulus "Min-Split Modulus", for which $x^n + 1$ can only be factorized into two polynomials with the degree of $n/2$.

It has been argued that less algebraic structure reduces the attacking surface [13]. In that spirit, and also for the sake of simplicity, we choose the min-split modulus $q = 251$ for our scheme.

Remark 2. Our selection principle is simply to minimize algebraic structures. Nonetheless, we do not see any weakness of the power of two modulus or the max-split modulus. In fact, it has been shown in theory [55] that Ring-LWE is hard for any ring of integers, which implies that $\mathbb{Z}_{2^\ell}/(x^n + 1)$ is as hard as any other choices, asymptotically speaking. Further, one can convert instances over one ring to another via modulus switching [6,8], at a cost of increased secrets and/or errors. In the meantime, from the implementation point of view, the modulus 257 and 256 may deliver better efficiency. We leave the concrete security of those types of modulus to future research.

4.2 Error Distribution.

In literatures, there are mainly two families of distributions that satisfy the average/worst case reduction theorem [57,46], namely, discrete Gaussian distribution [46,10] and centered binomial distribution [17]. Gaussian distribution consumes lots of entropy, is hard to implement (in constant time), and is also vulnerable to memory based side channel attacks [20] when implemented with look-up tables [30]. Therefore, we opt to use the centered binomial distribution for our scheme.

The next step is to determine the right parameters for the centered binomial distribution. Recall that the hardness of Ring-LWE problem is mainly determined by the dimension n and the error-modulus-ratio. When a byte-level modulus is used, the error-modulus-ratio becomes large enough even for small error

distributions. This allow us to use the simplest centered binomial distribution with $\lambda = 1$ as our basic error distribution.

In the implementation, as described in [10], a centered binomial distribution with the standard deviation of $\sqrt{\lambda/2}$ can be generated as $\sum_{i=1}^{\lambda}(b_i - \hat{b}_i)$, where $b_i, \hat{b}_i \in \{0, 1\}$ are uniformly random bits. That is, in order to get a centered binomial distribution with $\lambda = 1$, each element of the error vector is generated by $b - \hat{b}$, where $b, \hat{b}$ are uniformly random bits. More specifically, it will be a ternary distribution with $\Pr[x = -1] = 1/4$, $\Pr[x = 0] = 1/2$ and $\Pr[x = 1] = 1/4$. Note that, this distribution is formally defined as Ψ_1 in Section 2.2.

Remark 3. We wish to stress again that for such a small error distribution, the worst-case hardness of the Ring-LWE problem [46,55] will no longer hold. While this is a common approach in almost all lattice based cryptography [1], we still need to take a deep dive into the concrete security of the proposed parameters. The details will be presented in Section 5.

4.3 Decryption Errors

As shown in the decryption algorithm, the message is recovered via two steps. First, the error correction code word $\widehat{\boldsymbol{m}}$ is recovered from the ciphertext. Then, the message $\boldsymbol{m}$ is recovered from the code word. It is easy to verify that:

$$\begin{aligned}\widetilde{\boldsymbol{m}} &= \boldsymbol{c}_2 - (\boldsymbol{c}_1\boldsymbol{s})_{l_v} \\ &= (\boldsymbol{b}\boldsymbol{r})_{l_v} + \boldsymbol{e}_2 + \lfloor \tfrac{q}{2} \rceil \widehat{\boldsymbol{m}} - (\boldsymbol{c}_1\boldsymbol{s})_{l_v} \\ &= ((\boldsymbol{a}\boldsymbol{s} + \boldsymbol{e})\boldsymbol{r})_{l_v} + \boldsymbol{e}_2 + \lfloor \tfrac{q}{2} \rceil \widehat{\boldsymbol{m}} - ((\boldsymbol{a}\boldsymbol{r} + \boldsymbol{e}_1)\boldsymbol{s})_{l_v} \\ &= (\boldsymbol{e}\boldsymbol{r} - \boldsymbol{e}_1\boldsymbol{s})_{l_v} + \boldsymbol{e}_2 + \lfloor \tfrac{q}{2} \rceil \widehat{\boldsymbol{m}}\end{aligned} \tag{1}$$

Let $\boldsymbol{w} = (\boldsymbol{e}\boldsymbol{r} - \boldsymbol{e}_1\boldsymbol{s})_{l_v} + \boldsymbol{e}_2$, we have that the error rate of each $\widetilde{m}_i$ is $\delta = 1 - \Pr[-\lfloor \frac{q}{4} \rceil < w_i < \lfloor \frac{q}{4} \rceil]$. If $\boldsymbol{s}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{e}_1, \boldsymbol{e}_2$ are all randomly chosen from a small distribution with a standard deviation of σ and an expectation of 0, then according to the central limit theory, w_i follows a distribution that is very close to a discrete Gaussian distribution with a standard deviation of $\sigma^2\sqrt{2n}$ and an expectation of 0. Thus, the error rate for the each bit can be approximated by the Gaussian error function as $\delta \approx 1 - \mathsf{erf}\left(\frac{\lfloor q/4 \rceil}{\sqrt{2}(\sigma^2\sqrt{2n})}\right)$. For example, For $n = 512, q = 251$, and a distribution of Ψ_1 with a standard deviation $\sigma = 1/\sqrt{2}$, the error rate of each bit is estimated by:

$$\delta \approx 1 - \mathsf{erf}\left(\frac{\lfloor 251/4 \rceil}{\sqrt{2}((1/\sqrt{2})^2\sqrt{2 \times 512})}\right) \approx 2^{-13.195}.$$

Remark 4. This analysis assumes that each coefficient is linearly independent from another. Looking ahead, we will be applying error correction codes to reduce error rates. This functionality requires that the original codes are independent. From an information theoretical point of view this is not the case. It is easy to see that the coefficients are weakly correlated. Quantify the corrolations after ring multiplications has been an open problem for NTRU for many years [40]. As pointed out in [39,60], since the correlation is so weak, it is safe to treat the coefficients as if they were independent from each other.

Suppose the BCH code can correct l_t errors at most and the code word length is $l_n = l_v$, and assume the coefficients of $\boldsymbol{w}$ are independent from each other, we have the decryption error rate for a message $\boldsymbol{m}$:

$$\Delta \approx \sum_{j=l_t+1}^{l_v} \left(\binom{l_v}{j} \delta^j (1-\delta)^{l_v - j} \right) \tag{2}$$

Two factors influence the exact decryption error rate. First, the compress operation in the implementation will bring new errors. Briefly, in order to achieve best compactness, some bits of $\boldsymbol{c}$ are not included in the final ciphertext. This introduces errors that are computational indistinguishable from uniform, under the Ring-LWE assumption. Secondly, looking ahead, we will be relying on D2 error correction code [10] (in conjunction with BCH code) to minimize decryption fail rate for LAC-256v2. The message will be D2 encoded after going through the BCH process. Detailed estimation will be presented in Section 4.5.

4.4 Error Correction Code.

Our byte level modulus incurs a very high decryption error rate by design. Trivial or light error correction methods such as D2 or D4 code [10] are not capable of handling such a situation. Heavy error correction methods ought to be used for our use case. In the field of code theory, there are many powerful codes such as BCH, Goppa, LDPC, Turbo and Polar. In principle, any code with enough error correcting capability can be used in our scheme. For the sake of simplicity and efficiency we choose BCH code for implementation and benchmarking.

For BCH code with a code length l_n and a designed distance $l_d < l_n$, we use "`codes.BCHCode(GF(2),` l_n`,` l_d`)`" from SageMath [62] to obtain the concrete parameters. For example, for $l_n = 511$, $l_d = 41$, and a message size of $l_m = 340$, we get the parameter $[511, 340, 41]$ which says that the message can tolerant $l_t = (l_d - 1)/2 = 20$ errors at most. When $\delta = 2^{-13}$, we achieve a decryption error rate of $\Delta = 2^{-154}$ using equation (2).

Remark 5. Since the encoded message is encrypted as a Ring-LWE error element, an attacker does not see the structure of the code without breaking the Ring-LWE scheme in the first place. Therefore, the code structure will not affect the theoretical security of our scheme. Nonetheless, some code may not have a constant time implementation, and it is possible that error correction information may be partially leaked via side channels. To the best of our knowledge, we are not aware of any attack that quantifies or exploits such leakages.

4.5 Recommended Parameter Categories.

Having presented our design principles, we are ready to proceed with concrete parameter choices. We recommend the following parameter sets in Table 2, with respect to three categories of NIST post-quantum standardization project [1], namely, the equivalent security level of AES128, AES192 and AES256.

Concretely, dimensions $n = 512$ and $n = 1024$ with a basic error distribution Ψ_1 discussed as above are for the low security level LAC-128v2 and the high security level LAC-256v2, respectively. To get the middle security level LAC-192v2, we use a smaller secret and noise distribution $\Psi_{1/2}$ (defined in 2.2) and dimension 1024.

Note that it is sufficient to set the message size according to the security level, since in practice, public key encryption schemes are mainly used to encrypt session keys for symmetric encryption scheme. For the sake of simplicity, we set the message size to 256 for all security levels. In the previous version of LAC parameter sets that are submitted to NIST, the message size was twice as the security level.

The parameters of BCH code are selected to achieve a suitable decryption error rate and a high efficiency while defeating the high Hamming weight attacks (See section 5.2 for detail). We have $l_m = 256 + 8 = 264$ in our setting. The first 256 bits are used to store the message. In case where message is less than 256 bits we will utilize padding mechanism. This requires us to reserve an additional 8 bits to store the (real) message length. Next, for $l_m = 264$, the minimum available BCH code length l_n is 511. Lastly, we choose $l_d = 41$ which allows us to correct 20 bits of errors at most. The redundant data due to this error correction code is 23 bytes. To have a unified design across three parameter sets, we choose the same parameter, $[511, 264, 41]$, for all three security levels.

Note that the error rate for each coefficient is estimated by a convolution of all the error terms. In order to minimize the size of the ciphertext, in our implementation the lower 4 bits for each coefficient in $\boldsymbol{c}_2$ are rounded. This brings an additional uniformly random (under Ring-LWE assumption) error over $[-7, 7]$.

A public key consists of a 32 bytes seed $\mathsf{seed}_{\boldsymbol{a}}$, and an n bytes vector $\boldsymbol{b}$. A secret key is an n bytes vector[3]. In the case where Fujisaki-Okamoto transformation is used to achieve chosen ciphertext security, a secret key also contains a copy of the corresponding public key, so that the decryption algorithm can re-encrypt to check the validity of the ciphertext. Thus the size of a secret key becomes $2n + 32$ as shown in Table 1. Finally, a ciphertext contains both an n bytes vector $\boldsymbol{c}_1$, and l_v number of "half-byte" from $\boldsymbol{c}_2$ (since the lower 4 bits of each coefficient in $\boldsymbol{c}_2$ are rounded). For both LAC-128v2 and LAC-192v2 parameter sets, $l_v = l_m + 23 \times 8$, where 23 is the size of the redundant data. For LAC-256v2, $l_v = (l_m + 23 \times 8) \times 2$ due to the use of D2 encoding.

5 Concrete Security

We consider the best known generic attacks against Ring-LWE with our parameters, which treat the Ring-LWE problems as plain LWE problems. Those attacks are well-known by the community; their costs are well understood.

[3] One may simply store a 32 bytes seed for the secret key to minimize storage, at a cost of slightly slower decryption.

Categories	n	q	**dis**	**ecc**	**ml**	**pk**	**sk**	**ct**	**bit-er**	**dec-er**
LAC-128v2	512	251	Ψ_1	$[511, 264, 41]$	256	544	512	736	$2^{-12.42}$	2^{-146}
LAC-192v2	1024	251	$\Psi_{\frac{1}{2}}$	$[511, 264, 41]$	256	1056	1024	1248	$2^{-21.91}$	2^{-352}
LAC-256v2	1024	251	Ψ_1	[511,264,41]+D2	256	1056	1024	1472	$2^{-12.96}$	2^{-157}

dis secret and noise distributions **ecc** error correction code
ml message length **sk** secret key
pk public key **ct** ciphertext
bit-er single bit error rate without BCH **dec-er** decryption error rate.

Table 2. Recommended parameter of LAC

We also consider dedicated attacks that target specific designs of our scheme, namely the subfield attacks and the high Hamming weight attacks. Those attacks are firstly reported as comments to LAC submission to NIST-PQC. We will show that none of those dedicated attack works better than generic attacks for our (revised) parameter sets. Therefore, it is sufficient to use common methods (e.g. BKZ with (quantum) sieving) to evaluate the security of our scheme.

5.1 Generic Attacks

There are many generic algorithms to solve the LWE problem, see [8,63] for a survey of known techniques. It has been shown that lattice reduction attacks utilizing the BKZ algorithm [26] are more powerful than exhaustive search, combinational and algebraic algorithms. For simplicity, following the analysis of [9], we focus primly on two embedding attacks that are commonly referred to as primal attack and dual attack. We summarize the security estimates of both attacks in Table 3.

Algorithm	**Primal Attack**			**Dual Attack**		
	Classic	**Quantum**	**Block Size**	**Classic**	**Quantum**	**Block Size**
LAC-128v2	148	135	509	147	133	505
LAC-192v2	288	261	986	286	259	978
LAC-256v2	323	293	1105	320	290	1095

Classic: Classical complexity **Quantum:** Quantum complexity

Table 3. Concrete security of LAC

Primal attack. In a primal attack, one builds a lattice with a unique-SVP instance from the LWE samples; then, uses BKZ algorithm to recover this unique shortest vector. In a nutshell, given an LWE instance $(\boldsymbol{A}, \boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e})$, $\boldsymbol{A} \in \mathbb{Z}_q^{m \times n}$, the target lattice of dimension $d = m + n + 1$ is constructed as

$$\Lambda_{\boldsymbol{A}} = \{\boldsymbol{x} \in \mathbb{Z}^{m+n+1} : (\boldsymbol{A}|\boldsymbol{I}_m| - \boldsymbol{b})\boldsymbol{x} = \boldsymbol{0} \mod q\}.$$

It is easy to verify that, $\boldsymbol{v} = (\boldsymbol{s}, \boldsymbol{e}, 1)$ is the unique-SVP solution when both $\boldsymbol{s}$ and $\boldsymbol{e}$ are reasonably short. For exmaple, as shown in [9], the attack is successful if and only if $\sigma\sqrt{b} \leq \delta^{2b-d-1} \times q^{m/d}$, where σ is the standard deviation of the errors and secrets, $\delta = ((\pi b)^{1/b} b/2\pi e)^{1/(2(b-1))}$.

BKZ algorithm progressively processes the lattice basis by calling polynomial times a subroutine, such as the (quantum) sieving algorithm, to solve the exact shortest vector problem for sub-lattices with dimension (i.e. blocksize) b. This method is known as BKZ-core-(Q)Sieving, and its complexity depends solely on the block dimension b that is required for the BKZ algorithm to find the unique solution. According to [9], the best complexity of the SVP oracle is $\sqrt{3/2}^{b+o(b)} \approx 2^{0.292b}$ for classical sieving algorithms, and $\sqrt{13/9}^{b+o(b)} \approx 2^{0.265b}$ for quantum sieving algorithms.

Dual attack. In a dual attack, one firstly tries to build a dual lattice of the aforementioned primal lattice, and then uses the dual lattice to solve the decisional LWE problem. At a high level, given the LWE instance $(\boldsymbol{A}, \boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e})$, $\boldsymbol{A} \in \mathbb{Z}_q^{m \times n}$, the target lattice of dimension $d = m + n$ is constructed as

$$\Lambda_{\boldsymbol{A}}^{\perp} = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{Z}^m \times \mathbb{Z}^n : \boldsymbol{A}^t \boldsymbol{x} = \boldsymbol{y} \mod q\}.$$

Again, [9] showed that BKZ is capable of finding a vector $\boldsymbol{v} = (\boldsymbol{x}, \boldsymbol{y})$ of length $l = \delta^{d-1} q^{n/d}$, where the distance between $\boldsymbol{v}^t \boldsymbol{b}$ and the uniform distribution will be bounded by $\epsilon = 4\exp(-2\pi^2\tau^2)$ for $\tau = l\sigma/q$. This breaks the decisional LWE problem with an advantage ϵ.

Similar to primal attacks, the concrete security of dual attack also depends on the complexity of BKZ algorithm. There is a slight caveat when BKZ-core-QSieving is used: the attacker is able to amplify ϵ to $1/2$ by repeating the sieving algorithm for $R = max(1, 1/(\gamma\epsilon^2))$ times. This operation is almost free to the attacker, since sieving algorithm will produce $\gamma = 2^{0.2075b}$ vectors which far exceed the required number of short vectors $1/\epsilon^2$ for repeating.

Security Estimates. We use BKZ simulator with core-(Q)sieving model to estimate the security for our scheme. The required blocksize to achieve our target root Hermite factor is shown in Table 3. The corresponding security is then estimated for the obtained blocksize. Note that in [5], Albrecht *et al.* independently evaluated the security for all (Ring-)LWE candidates, and their estimation matches ours for LAC.

5.2 Dedicated Attacks

We stress again that the two attacks we are about to discuss does not perform better than generic attacks. Specifically, although we revised the parameters partially due to the threat of high Hamming weight attack, such a revision is only for conservative purpose and the attack itself does not work on both the original parameter sets and the revised ones.

Subfield Attacks. The idea of exploiting subfields is known to the lattice community for years [12,4,14,43], and to use this idea to analyze LAC was firstly proposed by Alperin-Sheriff [51] during the first round evaluation of NIST-PQC. Recall that $x^n + 1$ has two factors modulo $q = 251$:

$$x^n + 1 = (x^{n/2} + 91x^{n/4} + 250)(x^{n/2} + 160x^{n/4} + 250).$$

In other words, there exist two subfields defined by two polynomials $\boldsymbol{g}$ and $\boldsymbol{h}$ where $\boldsymbol{g} = x^{n/2} + 91x^{n/4} + 250$ and $\boldsymbol{h} = x^{n/2} + 160x^{n/4} + 250$.

Given $(\boldsymbol{a}, \boldsymbol{b} = \boldsymbol{as} + \boldsymbol{e})$, one may recover $(\boldsymbol{s}, \boldsymbol{e})$ by looking at the samples over the subfields. It may be sufficient to recover $(\boldsymbol{s}_g := \boldsymbol{s} \bmod \boldsymbol{g}, \boldsymbol{e}_g := \boldsymbol{e} \bmod \boldsymbol{g})$ from $(\boldsymbol{a} \bmod \boldsymbol{g}, \boldsymbol{b} \bmod \boldsymbol{g})$, and $(\boldsymbol{s}_h, \boldsymbol{e}_h$, respectively). Next, it becomes straightforward to recover $(\boldsymbol{s}, \boldsymbol{e})$ via Chinese remainder theorem.

Analysis. In the rest, we give a full analysis of this attack. The key point of the attack is that by moving to the subfield, the lattice dimension is practically halved. Therefore, the BKZ complexity may be reduced for the new sub-lattices. Note that this is not necessarily always the case under core-(Q)Sieving model where only the cost of subroutine counts; and the cost of the subroutine depends only on the root Hermite factor. Nonetheless, to have a meaningful analysis, we assume that this is not an obstacle: the attacker may access an SVP oracle for BKZ subroutines solely for this attack.

Our analysis will show that the corresponding vectors in the subfields, $(\boldsymbol{s}_g, \boldsymbol{e}_g)$, will be larger than the Gaussian heuristic length. In other words, even if one were able to perform lattice reduction over the dimension-halved lattices, they will not be able to recover the desired vectors.

The attack reduces the dimension, in the meantime, the modulo operation increases the size of $(\boldsymbol{s}_g, \boldsymbol{e}_g)$ (similarly, $(\boldsymbol{s}_h, \boldsymbol{e}_h)$). To be precise, when $(\boldsymbol{s}, \boldsymbol{e})$ are small polynomials with the coefficients in $\{-1, 0, 1\}$, the coefficients of $(\boldsymbol{s}_g, \boldsymbol{e}_g)$ will lie in $\{0, \pm 1, \pm 2, \pm 91\}$. Coefficients of ± 91 will be too large. Alperin-Sheriff also pointed out that by multiplying $\boldsymbol{s}$ and $\boldsymbol{e}$ by 11, all the coefficients of $(\boldsymbol{s}_g, \boldsymbol{e}_g)$ will be within the interval of $[-25, 25]$.

Let $\boldsymbol{A} = [\boldsymbol{A}_g | \boldsymbol{I} | 11 \times \boldsymbol{b}_g]$, where $\boldsymbol{A_g}$ denotes the matrix generated by $\boldsymbol{a}_g$, if $\boldsymbol{z} = [11 \times \boldsymbol{s}_g | 11 \times \boldsymbol{e}_g | -1]$ is the shortest solution of $\boldsymbol{Az} = 0 \bmod q$, we can recover $\boldsymbol{z}$ with the primal attack. Note that, the dimension of a primal attack is reduced from $d = 2n + 1$ to $d = n + 1$ via the subfield attack. Since $\boldsymbol{A}$ is a random matrix, the q-ary lattice $\Lambda_q^{\perp}(\boldsymbol{A})$ will behave as a random lattice [28], and therefore it is sufficient to use Gaussian heuristic to estimate the length of shortest vectors in this lattice:

$$\lambda_1(\Lambda_q^{\perp}) \approx q^{m/d} \sqrt{\frac{d}{2\pi e}}.$$

In the case of $n = 512$ and $n = 1024$, the lengths of the shortest vector is expected at 86.36 and 122.4, respectively.

On the other hand, we also need to estimate the length of $\boldsymbol{z}$. Central limit theory says that the length of $\boldsymbol{z}$ approximately follows a discrete Gaussian distribution. Our implementation shows that $\boldsymbol{z}$ closely follows a Gaussian distribution

with a mean and deviation pair of $(253.59, 6.9)$ for LAC-128v2, $(253.26, 6.29)$ for LAC-192v2 and $(358.42, 6.86)$ for LAC-256v2[4].

It is easy to verify that, the length of $\boldsymbol{z}$ will be larger than the solution of $\boldsymbol{Az} = 0 \bmod q$ except for negligible probability. Hence $\boldsymbol{z}$ will not be a short vector in this lattice. In other words, if one were to use subfield attack, and assuming that they have free access to SVP oracles simply for the sub-lattices, they will not be able to locate the vector.

To sum up, the subfield attack described above will not affect the security of LAC for either original parameter sets or the revised version.

High Hamming Weight Attack. This is a chosen ciphertext attack that exploits the fact that the secrets and errors $(\boldsymbol{r}, \boldsymbol{e}_1)$ in some ciphertexts (with certain probability) may have higher-than-usual Hamming weight. It is feasible since $(\boldsymbol{r}, \boldsymbol{e}_1)$ are randomly selected from Ψ_1 or $\Psi_{\frac{1}{2}}$. It is easy to see that the decryption error rate is influenced by the Hamming weight. Therefore, with enough number of random samples, an attacker may obtain sufficient number of samples whose secrets and errors have very higher Hamming weight, and then invoke the decryption oracle to extract information of the private key.

Analysis. It has been shown that chosen plaintext secure version of (Ring-)LWE based schemes suffer from an reaction attack [33]. To address this vulnerability, most schemes rely on Fujisaki-Okamoto transformation [35,36,41] to achieve chosen ciphertext security. We also adopt the same approach. Via this transformation, the randomness vectors $(\boldsymbol{r}, \boldsymbol{e}_1)$ are generated from the plaintext message by a pseudorandom generator. Thus the vectors $(\boldsymbol{r}, \boldsymbol{e}_1)$ are randomly distributed from the view of the adversary.

In a comment to LAC [1], Alperin-Sheriff showed that, for the LAC-256v1 parameter set, the probability that a pair of valid $(\boldsymbol{r}, \boldsymbol{e}_1)$ with a Hamming weight of at least $1024 + 310 = 1334$ is greater than

$$\binom{2048}{1334} / 2^{2048} = 2^{-143}.$$

Therefore, with 2^{207} pre-computations (assuming each access to the pseudorandom generator incurs a cost of 1), the adversary will obtain 2^{64} messages for which the corresponding $(\boldsymbol{r}, \boldsymbol{e}_1)$ have Hamming weight exceeding 1334. It is worth noting that the adversary only needs to access the decryption oracle for 2^{64} times in this setting. Next, for samples with such high Hamming weights, the decryption error rate for each bit of $\widetilde{m}_i$ is expected at

$$\delta_{high} \approx 1 - \mathsf{erf}\left(\frac{\lfloor 251/4 \rceil}{\sqrt{2}((1/\sqrt{2})\sqrt{(1024+310)/2048}\sqrt{2 \times 1024})}\right) \approx 2^{-5.9},$$

[4] The data is obtained over 100,000 random samples for each parameter set using SageMath. The experiment is not mean to extensive to show any proof of statistical distances; the mean is obviously much higher than Gaussian heuristic length.

This yields a decryption error rate for the message $\boldsymbol{m}$:

$$\Delta_{high} = \sum_{j=55+1}^{1023} \left(\binom{1023}{j} \delta_{high}^{j} (1-\delta_{high})^{1023-j} \right) \approx 2^{-44.4}.$$

As a result, with 2^{207} pre-computations and 2^{64} decryption oracle queries, the adversary can get about $2^{19.6}$ decryption failures. We remark that, 1334 is a lower bound of the Hamming weight for decryption errors. Decryption errors may occur for any Hamming weight above 1334, and therefore the adversary may get (a little) more than $2^{19.6}$ decryption failures if they were to perform all above (pre-)computations.

Remark 6. We argue that, as also pointed by D'Anvers [1], it is difficult to get any information about the private key from these decryption failures. All the information that an adversary may learn is whether there are more than l_t errors in the code word; they cannot determine which coefficients are failing as in a reaction attack [33]. Nevertheless, we do recognize this as a leakage of information and present our counter-measures.

Counter-measures. Our objective is to make LAC completely immune to such attack by decreasing the decryption error rate. Our method, at a high level, is to reduce the message length from 512 bits (as in the original parameter sets LAC-256v1 submitted to NIST-PQC) to 256 bits. In doing so, we allow more space for error corrections and therefore achieve an even smaller decryption error rate.

Following the above example, with a message size of 256, the BCH code can correct up to $l_t = 100$ errors for the code length of 1023. Consequently, the decryption error rate for high Hamming weight random vectors $\boldsymbol{r}, \boldsymbol{e}_1$ is estimated as:

$$\Delta_{high} = \sum_{j=100+1}^{1023} \left(\binom{1023}{j} \delta_{high}^{j} (1-\delta_{high})^{1023-j} \right) \approx 2^{-147}.$$

As a result, with 2^{64} message queries, the probability that the adversary gets one decryption failure is around 2^{-83}. In other words, it will take the adversary over 2^{256} operations to get a single decryption error.

However, we notice that, when l_t is greater than 50 the decoding efficiency decreases drastically. To resolve this problem, for LAC-256v2 we use the D2 error correction code [16,9] together with the BCH code. That is, the message is firstly encoded with BCH, then the code word is encoded with D2. As a result, the BCH code only need to correct less than 20 errors.

The upper bound of the decryption error rate of our recommended parameters in the case of high Hamming weight attack is presented as follow. We give the upper bound the Hamming weight that the adversary can obtain after 2^l operations of pre-computation, where l is the security level. Then we estimate the bit error rate and decryption error rate according to this upper bound of Hamming weight. It is clear that, for each parameter set, the decryption failure occurs with a negligible probability in the security parameter.

Categories	Ham$(\boldsymbol{r}, \boldsymbol{e}_1)$	Prob	Bit Error Rate	Error Rate
LAC-128v2	512+206	2^{-128}	$2^{-9.59}$	2^{-87}
LAC-192v2	512+333	2^{-192}	$2^{-14.75}$	2^{-201}
LAC-256v2	1024+416	2^{-256}	$2^{-9.77}$	2^{-90}

Ham$(\boldsymbol{r}, \boldsymbol{e}_1)$ denotes the Hamming weight of $(\boldsymbol{r}, \boldsymbol{e}_1)$, Prob denotes the probability that the adversary obtains $(\boldsymbol{r}, \boldsymbol{e}_1)$ with target Hamming weight in pre-computation.

Table 4. Decryption error rate of high Hamming attack

6 Implementation

As mentioned earlier, an important difference between LAC and previous Ring-LWE based public key encryption schemes is that our parameters do not support NTT. In this section, we present some highlights of our customized implementation, including a generalized polynomial multiplication method (as per NIST-PQC's request) and an optimized version based on AVX2 instructions. The source code of our scheme is available at https://github.com/luxianhui007/LAC.

6.1 Polynomial Multiplication

Polynomial multiplication is the most time consuming operation in the implementation of LAC. In addition to a reference implementation, we provide two optimized versions as follows:

- **General Optimized Version:** Our main observation is that, since $\mathbf{s}$ and $\mathbf{r}$ are selected from $\{-1, 0, 1\}$, the multiplication operation can be implemented by bitwise logical `AND` operation as $a_i \times 1 = a_i$&0xff and $a_i \times 0 = a_i$&0x00. Further more, we can pack 4 items into one `uint64_t` data type. Then, polynomial multiplication becomes simply $\mathbf{as} = \sum_{s_i=1} a_i - \sum_{s_i=-1} a_i$.

  Remark 7. With $q < 256$, it is possible to pack 8 coefficients into a single `uint64_t` unit, in theory. We choose to hold 4 coefficients at a time, and use the free space as a buffer for the carriers, so that we are not obliged to perform mod reductions after every arithmetic operation. This yields better performance in practice.

- **AVX2 Based Version:** AVX2 allows us to handle 256 bits data type. We are able to store 32 coefficients in a single `__mm256` data type, and utilize `_mm256_maddubs_epi16` instruction which does 32 multiplications and adjacent addition operation in a single operation. We obtain approximately 30x acceleration with this optimization.

6.2 Benchmark

In this section we present benchmark results of the chosen plaintext secure version of LAC. The test bed sits on an ubuntu 16.04 operation system with an

Intel Core-i7-4770S (Haswell) @ 3.10GHz processor and 7.6GB of memory, with Turbo Boost and Hyperthreading disabled.

Categories	Key generation		Encryption		Decryption	
	CPU Cycles	Times(μs)	CPU Cycles	Times(μs)	CPU Cycles	Times(μs)
LAC-128v2	105,193	37	192,777	58	81,539	29
LAC-192v2	354,313	114	498,461	162	293,976	93
LAC-256v2	325,752	106	604,737	199	292,497	95

Table 5. Performance of general optimized version of LAC

Categories	Key generation		Encryption		Decryption	
	CPU Cycles	Times(μs)	CPU Cycles	Times(μs)	CPU Cycles	Times(μs)
LAC-128v2	42,506	13	64,819	20	24,654	7
LAC-192v2	110,199	36	178,961	46	68,637	21
LAC-256v2	99,516	31	164,560	52	72,888	22

Table 6. Performance of AVX2 version of LAC

7 Conclusion and future work

We propose the first instantiation of Ring-LWE based public key encryption scheme with byte-level modulus. Our main contributions include the selection of suitable byte-level modulus, security evaluations of special attacks according to the our concrete parameters, and efficient implementations without NTT. Compared with existing schemes, our new scheme is more compact and nearly as efficient as NewHope and Kyber.

Side Channel Resistance. Side channel resistance is an important notion for implementations. The common side channels are timing leakage and memory leakage. Most NIST-PQC candidates provides constant time implementation. The current LAC implementation is not constant time since the number of none-zero coefficients in the secret is not a constant; the Hamming weight of the secret is leaked through the timing data. To achieve a constant time multiplication, one may either use a fixed Hamming weight ternary distribution, or replace the index based algorithm with Karatsuba or Toom-Cook algorithms. The first modification requires a full review of parameters, and the secret distributions align less with theoretical reductions; while the second one decreases performances. We leave this research to future work.

Power-of-two Modulus. Power-of-two modulus offers the same security from a theoretical point of view. They are even more efficient than our current choice

in practice since modulus reduction is simply omitting the higher bits. We leave it as a future work to better understand the underlying structure and its impact to concrete security.

References

1. Nist post-quantum cryptography project. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions
2. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings. pp. 553–572 (2010), https://doi.org/10.1007/978-3-642-13190-5_28
3. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings. pp. 98–115 (2010), https://doi.org/10.1007/978-3-642-14623-7_6
4. Albrecht, M.R., Bai, S., Ducas, L.: A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes. In: Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I. pp. 153–178 (2016), https://doi.org/10.1007/978-3-662-53018-4_6
5. Albrecht, M.R., Curtis, B.R., Deo, A., Davidson, A., Player, R., Postlethwaite, E.W., Virdia, F., Wunderer, T.: Estimate all the {LWE, NTRU} schemes! In: Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings. pp. 351–367 (2018), https://doi.org/10.1007/978-3-319-98113-0_19
6. Albrecht, M.R., Faugère, J., Fitzpatrick, R., Perret, L.: Lazy modulus switching for the BKW algorithm on LWE. In: Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings. pp. 429–445 (2014), https://doi.org/10.1007/978-3-642-54631-0_25
7. Albrecht, M.R., Göpfert, F., Virdia, F., Wunderer, T.: Revisiting the expected cost of solving usvp and applications to LWE. In: Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I. pp. 297–322 (2017), https://doi.org/10.1007/978-3-319-70694-8_11
8. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. J. Mathematical Cryptology 9(3), 169–203 (2015), http://www.degruyter.com/view/j/jmc.2015.9.issue-3/jmc-2015-0016/jmc-2015-0016.xml
9. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Newhope without reconciliation. IACR Cryptology ePrint Archive 2016, 1157 (2016), http://eprint.iacr.org/2016/1157
10. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange - A new hope. In: 25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016. pp. 327–343 (2016), https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim

11. Baan, H., Bhattacharya, S., García-Morchón, Ó., Rietman, R., Tolhuizen, L., Torre-Arce, J.L., Zhang, Z.: Round2: KEM and PKE based on GLWR. IACR Cryptology ePrint Archive 2017, 1183 (2017), http://eprint.iacr.org/2017/1183
12. Bernstein, D.: A subfield-logarithm attack against ideal lattices (2014), https://blog.cr.yp.to/20140213-ideal.html
13. Bernstein, D.J., Chuengsatiansup, C., Lange, T., van Vredendaal, C.: NTRU prime: Reducing attack surface at low cost. In: Selected Areas in Cryptography - SAC 2017 - 24th International Conference, Ottawa, ON, Canada, August 16-18, 2017, Revised Selected Papers. pp. 235–260 (2017), https://doi.org/10.1007/978-3-319-72565-9_12
14. Biasse, J., Espitau, T., Fouque, P., Gélin, A., Kirchner, P.: Computing generator in cyclotomic integer rings - A subfield algorithm for the principal ideal problem in $l_{|\Delta\urcorner|}(\frac{1}{2})$ and application to the cryptanalysis of a FHE scheme. In: Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I. pp. 60–88 (2017), https://doi.org/10.1007/978-3-319-56620-7_3
15. Bos, J.W., Costello, C., Ducas, L., Mironov, I., Naehrig, M., Nikolaenko, V., Raghunathan, A., Stebila, D.: Frodo: Take off the ring! practical, quantum-secure key exchange from LWE. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016. pp. 1006–1018 (2016), http://doi.acm.org/10.1145/2976749.2978425
16. Bos, J.W., Costello, C., Naehrig, M., Stebila, D.: Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In: 2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015. pp. 553–570 (2015), https://doi.org/10.1109/SP.2015.40
17. Bos, J.W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Stehlé, D.: Crystals-kyber: a cca-secure module-lattice-based kem. IACR Cryptology ePrint Archive 2017, 634 (2017)
18. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. In: Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012. pp. 309–325 (2012), http://doi.acm.org/10.1145/2090236.2090262
19. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011. pp. 97–106 (2011), https://doi.org/10.1109/FOCS.2011.12
20. Bruinderink, L.G., Hülsing, A., Lange, T., Yarom, Y.: Flush, gauss, and reload - A cache attack on the BLISS lattice-based signature scheme. In: Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings. pp. 323–345 (2016), https://doi.org/10.1007/978-3-662-53140-2_16
21. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings. pp. 523–552 (2010), https://doi.org/10.1007/978-3-642-13190-5_27
22. Castryck, W., Iliashenko, I., Vercauteren, F.: Provably weak instances of ring-lwe revisited. In: Advances in Cryptology - EUROCRYPT 2016 - 35th Annual

International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I. pp. 147–167 (2016), https://doi.org/10.1007/978-3-662-49890-3_6
23. Chen, H., Lauter, K.E., Stange, K.E.: Attacks on search RLWE. vol. 2015, p. 971 (2015), http://eprint.iacr.org/2015/971
24. Chen, H., Lauter, K.E., Stange, K.E.: Vulnerable galois RLWE families and improved attacks. vol. 2016, p. 193 (2016), http://eprint.iacr.org/2016/193
25. Chen, L., Jordan, S., Liu, Y.K., Moody, D., Peralta, R., Perlner, R., Smith-Tone, D.: Report on post-quantum cryptography. Tech. rep. (2016), https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf
26. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings. pp. 1–20 (2011), https://doi.org/10.1007/978-3-642-25385-0_1
27. Dai, W., Whyte, W., Zhang, Z.: Optimizing polynomial convolution for ntruencrypt. IACR Cryptology ePrint Archive 2018, 229 (2018), http://eprint.iacr.org/2018/229
28. Daniele Micciancio, O.R.: Lattice-based cryptography. Tech. rep., https://cims.nyu.edu/ regev/papers/pqc.pdf (2008)
29. Ding, J.: A simple provably secure key exchange scheme based on the learning with errors problem. IACR Cryptology ePrint Archive 2012, 688 (2012), http://eprint.iacr.org/2012/688
30. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. In: Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. pp. 40–56 (2013), https://doi.org/10.1007/978-3-642-40041-4_3
31. Eisenträger, K., Hallgren, S., Lauter, K.E.: Weak instances of PLWE. In: Selected Areas in Cryptography - SAC 2014 - 21st International Conference, Montreal, QC, Canada, August 14-15, 2014, Revised Selected Papers. pp. 183–194 (2014), https://doi.org/10.1007/978-3-319-13051-4_11
32. Elias, Y., Lauter, K.E., Ozman, E., Stange, K.E.: Provably weak instances of ring-lwe. In: Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I. pp. 63–92 (2015), https://doi.org/10.1007/978-3-662-47989-6_4
33. Fluhrer, S.R.: Cryptanalysis of ring-lwe based key exchange with key share reuse. IACR Cryptology ePrint Archive 2016, 85 (2016), http://eprint.iacr.org/2016/085
34. Fritzmann, T., Pöppelmann, T., Sepúlveda, J.: Analysis of error-correcting codes for lattice-based key exchange. IACR Cryptology ePrint Archive 2018, 150 (2018), http://eprint.iacr.org/2018/150
35. Fujisaki, E., Okamoto, T.: How to enhance the security of public-key encryption at minimum cost. In: Public Key Cryptography, Second International Workshop on Practice and Theory in Public Key Cryptography, PKC '99, Kamakura, Japan, March 1-3, 1999, Proceedings. pp. 53–68 (1999), https://doi.org/10.1007/3-540-49162-7_5
36. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. J. Cryptology 26(1), 80–101 (2013), https://doi.org/10.1007/s00145-011-9114-1

37. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008. pp. 197–206 (2008), http://doi.acm.org/10.1145/1374376.1374407
38. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. pp. 75–92 (2013), https://doi.org/10.1007/978-3-642-40041-4_5
39. Hoffstein, J., Pipher, J., Schanck, J.M., Silverman, J.H., Whyte, W., Zhang, Z.: Choosing parameters for ntruencrypt. In: Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings. pp. 3–18 (2017), https://doi.org/10.1007/978-3-319-52153-4_1
40. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings. pp. 267–288 (1998), https://doi.org/10.1007/BFb0054868
41. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: Ind-cca-secure key encapsulation mechanism in the quantum random oracle model, revisited. In: Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III. pp. 96–125 (2018), https://doi.org/10.1007/978-3-319-96878-0_4
42. Jin, Z., Zhao, Y.: Optimal key consensus in presence of noise. CoRR abs/1611.06150 (2016), http://arxiv.org/abs/1611.06150
43. Kirchner, P., Fouque, P.: Revisiting lattice attacks on overstretched NTRU parameters. In: Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I. pp. 3–26 (2017), https://doi.org/10.1007/978-3-319-56620-7_1
44. Lindner, R., Peikert, C.: Better key sizes (and attacks) for lwe-based encryption. In: Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings. pp. 319–339 (2011), https://doi.org/10.1007/978-3-642-19074-2_21
45. Lyubashevsky, V., Micciancio, D.: Generalized compact knapsacks are collision resistant. In: Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II. pp. 144–155 (2006), https://doi.org/10.1007/11787006_13
46. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings. pp. 1–23 (2010), https://doi.org/10.1007/978-3-642-13190-5_1
47. Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-lwe cryptography. In: Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings. pp. 35–54 (2013), https://doi.org/10.1007/978-3-642-38348-9_3
48. Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In: 43rd Symposium on

Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings. pp. 356–365 (2002), https://doi.org/10.1109/SFCS.2002.1181960
49. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings. pp. 700–718 (2012), https://doi.org/10.1007/978-3-642-29011-4_41
50. Micciancio, D., Peikert, C.: Hardness of SIS and LWE with small parameters. In: Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. pp. 21–39 (2013), https://doi.org/10.1007/978-3-642-40041-4_2
51. NIST: NIST PQC FORUM: LAC, https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/official-comments/LAC-official-comment.pdf
52. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009. pp. 333–342 (2009), http://doi.acm.org/10.1145/1536414.1536461
53. Peikert, C.: Lattice cryptography for the internet. In: Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings. pp. 197–219 (2014), https://doi.org/10.1007/978-3-319-11659-4_12
54. Peikert, C.: How (not) to instantiate ring-lwe. In: Security and Cryptography for Networks - 10th International Conference, SCN 2016, Amalfi, Italy, August 31 - September 2, 2016, Proceedings. pp. 411–430 (2016), https://doi.org/10.1007/978-3-319-44618-9_22
55. Peikert, C., Regev, O., Stephens-Davidowitz, N.: Pseudorandomness of ring-lwe for any ring and modulus. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017. pp. 461–473 (2017), http://doi.acm.org/10.1145/3055399.3055489
56. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008. pp. 187–196 (2008), http://doi.acm.org/10.1145/1374376.1374406
57. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005. pp. 84–93 (2005), http://doi.acm.org/10.1145/1060590.1060603
58. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM 56(6), 34:1–34:40 (2009), http://doi.acm.org/10.1145/1568318.1568324
59. Rosca, M., Stehlé, D., Wallet, A.: On the ring-lwe and polynomial-lwe problems. In: Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I. pp. 146–173 (2018), https://doi.org/10.1007/978-3-319-78381-9_6
60. Saarinen, M.O.: HILA5: on reliability, reconciliation, and error correction for ring-lwe encryption. In: Selected Areas in Cryptography - SAC 2017 - 24th International Conference, Ottawa, ON, Canada, August 16-18, 2017, Revised Selected Papers. pp. 192–212 (2017), https://doi.org/10.1007/978-3-319-72565-9_10

61. Saarinen, M.J.O.: On reliability, reconciliation, and error correction in ring-lwe encryption. IACR Cryptology ePrint Archive 2017, 424 (2017), http://eprint.iacr.org/2012/688
62. Sage: SageMath, http://www.sagemath.org/
63. Schmidt, M., Bindel, N.: Estimation of the hardness of the learning with errors problem with a restricted number of samples. IACR Cryptology ePrint Archive 2017, 140 (2017), http://eprint.iacr.org/2017/140
64. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient public key encryption based on ideal lattices. In: Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings. pp. 617–635 (2009), https://doi.org/10.1007/978-3-642-10366-7_36