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], Bao Li[1,2,3], Kunpeng Wang[1,2,3],
Zhe Liu[5] and Hao Yang[5]

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.
5. Nanjing University of Aeronautics and Astronautics
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 push the limit of state-of-the-art, and propose the first instantiation to the family of ring learning with error based solutions where the modulus is at a byte level and the noise is at a bit level. Additionally, we present techniques to deal with side affects incurred by the decreasing of the modulus, namely, the increase of decryption errors, and the loss of efficiency. Our method of managing decryption errors has sparked rich discussion and new analysis within the post-quantum community. In the meantime, our choice of parameters, although are no longer NTT-friendly, still allows for high parallelization. We present implementation characteristics on Intel x86-64. Our result shows that LAC is more compact than all existing (Ring-) LWE based solutions, and only slightly less efficient compared with popular solutions in this domain, such as 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 26 candidates that enter the second round of NIST-PQC standardization process [2].

(NIST) [24,2]. 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. As a typical example 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 [60], and became extremely versatile in constructing public key encryption schemes [59,55,47,53,17], identity based encryption schemes [39,23,4,5] and fully homomorphic encryption schemes [21,20,40]. 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 [42] and the ring-based short integer solution problem [52,49], Lyubashevsky, Peikert and Regev [50,51,58] 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.* [65] also proposed a special case of Ring-LWE over power-of-two cyclotomic polynomials; in [58], 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 $O(n \log n)$ by using the fast Fourier transform (FFT) or number theoretic transform (NTT). 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 [18,12,11,19,45,63,13].

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 [60] and Lyubashevsky *et al.* [50]. For the key exchange protocols, one may use the reconciliation method, first put forth by Ding [32], and then refined by Peikert [56], to improve efficiency.

Subsequent works, such as [18,12,19,63,13], have contributed to a large portion of the NIST post-quantum cryptography standardization process (NIST-PQC) [2]. 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, 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 [34,12], to illustrate our idea, although we must remark that it is shown by Peikert *et al.* in [58] 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 [10,12,9], 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, in the spirit of trading time complexity for space complexity, it makes sense to choose the modulus as small as possible (and pay the efficiency penalty), while keeping the ratio somewhat a constant to maintain a same security level.

In this paper, we investigate the above approach. 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 [65,50,58,62]. 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 [25,57,7]. Arguably, most lattice based submissions to NIST-PQC follow a similar design [50,47,32,56], 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 [19], 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 analysis such as BKZ with (quantum) sieving [10,12,9] and hybrid attacks [43,41], as well as dedicated attacks, such as the subfield attacks and hamming weight attacks [54].

Error Corrections. In most lattice based schemes, dated back from one of the first lattice based encryption schemes, NTRU [42], 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 [36], 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 [12], do not work well in our use case.

To cope with this error growth, 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. With BCH code we are able to decrease the decryption error rate to a desirable level.

We remark that our usage of heavy error correction mechanism has sparked fruitful discussions which lead to advancement of cryptanalysis in the field, for instance, see [31,30,29,28]. Looking ahead, our revised parameter sets are robust against all those attacks. We also note that, the choice of error correction code will not affect the theoretical 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 and Comparison. 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 [15,26] and index based multiplications algorithms, such as [13]. 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 $\sigma = 1/\sqrt{2}$.

We propose optimized implementation of polynomial multiplication with vector instructions such as AVX2 over the Intel64. Our code is publicly available at [1]. The authors do not claim any patent toward the material of this paper or the related source code.

To highlight the compactness and efficiency of our scheme, we briefly compare the performance of our scheme with NewHope [11] and Kyber [19]. We compare the chosen ciphertext secure version of the schemes. All three schemes uses BKZ with (quantum) sieving to estimate their security. Note that our estimation is independently confirmed in [7].

In a nutshell, LAC outperforms both NewHope and Kyber at 128 and 256 bits security levels, in terms of the ciphertext size; while remaining adequately more generous on the security margin. As expected, LAC pays the cost by being less efficient due to the error correction code.

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	800	736	33,428	49,184	40,564	100
LAC-128	1056	544	712	59,584	89,055	140,221	133
Kyber768	2400	1184	1088	62,396	83,748	70,304	164
LAC-192	2080	1056	1188	119,246	137,653	320,135	259
NewHope1024	3680	1824	2208	129,670	210,092	220,864	233
Kyber1024	3168	1568	1568	88,568	115,952	99,764	230
LAC-256	2080	1056	1424	135,780	207,938	359,209	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

Parameters obtained from Round-2 submission to NIST-PQC [3]. Benchmark performed on Intel Core i7-4770 with Turbo Boost and Hyperthreading disabled.

Versions and Modifications. To the best of our knowledge, our new scheme is the first Ring-LWE based public key encryption scheme with byte-level modulus. An earlier version of our scheme, hereafter referred to as LACv1 was submitted to the first round of NIST-PQC. This paper summarizes our work of LACv1, as will as our new version LACv2. The major difference between the two versions includes the message length, secret and error distribution, and subsequently a different error correction parameter. These modifications better address several potential risks arised during the first round evaluation [54].

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 $\boldsymbol{a} = (a_0, a_1, ..., a_{m-1})$, its l_1-norm is defined as $\|\boldsymbol{a}\|_1 = \sum_{i=0}^{m-1} |a_i|$; the l_2-norm, also known as the Euclidean norm, is defined as $\|\boldsymbol{a}\|_2 = \sqrt{\sum_{i=0}^{m-1} a_i^2}$, or solely denoted as $\|\boldsymbol{a}\|$. 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, $x \overset{\$}{\leftarrow} S$ denotes that an element x is chosen from S uniformly at random. For a distribution D, $x \overset{\$}{\leftarrow} D$ denotes that a random variable x is sampled according to D. For a randomized algorithm A, $y \overset{\$}{\leftarrow} \mathsf{A}(x)$ 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 [12], in order to mitigate the heavy cost of Gaussian sampling. Let Ψ_σ be the centered binomial distribution with σ being the parameter of the distribution, where the corresponding standard variance is $\sqrt{\frac{\sigma}{2}}$. In the design of LAC we also use centered binomial distribution with parameters 1 and $\frac{1}{2}$ (denoted as Ψ_1 and $\Psi_{\frac{1}{2}}$ respectively) as follows:

Definition 1 **(Ψ_1).** *Sample $(a, b) \overset{\$}{\leftarrow} \{0, 1\}^2$, 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 $(a, b) \overset{\$}{\leftarrow} \Psi_1$, 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 $\frac{1}{4}$.*

For a positive integer n, Ψ_σ^n denotes the n independently identical distribution of Ψ_σ. When sampling according to Ψ_σ^n, the components of the random variable are all independently chosen.

Besides, we define n-ary centered binomial distribution with fixed Hamming weight, denoted as $\Psi_{\sigma}^{n,h}$, when $0 < h < n/2$ is even. For a random variable according to the distribution, its Hamming weight is fixed to the expectation h, and the numbers of both 1's and -1's are $h/2$, the number of 0 is $n-h$.

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 $x \overset{\$}{\leftarrow} D$. 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 $1 \le i \le t$.

2.3 Learning with Errors (over Rings)

We refer the readers to [60,61,65,50,58] 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 $\boldsymbol{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{m \times n}$, the secret $\boldsymbol{s} \overset{\$}{\leftarrow} \chi_{\boldsymbol{s}}^n$ and the error $\boldsymbol{e} \overset{\$}{\leftarrow} \chi_{\boldsymbol{e}}^m$.*

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 $\boldsymbol{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{m \times n}$, $\boldsymbol{s} \overset{\$}{\leftarrow} \chi_{\boldsymbol{s}}^n$, $\boldsymbol{e} \overset{\$}{\leftarrow} \chi_{\boldsymbol{e}}^m$, $\boldsymbol{u} \overset{\$}{\leftarrow} \mathbb{Z}_q^m$.*

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 $\boldsymbol{a} \overset{\$}{\leftarrow} \mathcal{R}_q$, the secret $\boldsymbol{s} \overset{\$}{\leftarrow} \chi_{\boldsymbol{s}}$ and the error $\boldsymbol{e} \overset{\$}{\leftarrow} \chi_{\boldsymbol{e}}$.*

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 $\boldsymbol{a} \overset{\$}{\leftarrow} \mathcal{R}_q$, $\boldsymbol{s} \overset{\$}{\leftarrow} \chi_{\boldsymbol{s}}$, $\boldsymbol{e} \overset{\$}{\leftarrow} \chi_{\boldsymbol{e}}$, $\boldsymbol{u} \overset{\$}{\leftarrow} \mathcal{R}_q$.*

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:

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

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 [50], 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

Notations. Let q be the modulus. Define the polynomial ring $R_q = \mathbb{Z}_q/(x^n+1)$.

Define the message space $\mathcal{M}$ be $\{0,1\}^{l_m}$ for a positive integer l_m, and the space of random seeds $\mathcal{S}$ be $\{0,1\}^{l_s}$ for a positive integer l_s. The integers l_m and l_s will be specified afterwards.

We use n independently identical distribution of Ψ_σ, namely Ψ_σ^n. Beside, we use the n-ary centered binomial distribution $\Psi_\sigma^{n,h}$, where the concrete choices of parameters will be given later.

Subroutines. In the subroutines dealing with the encoding and decoding of the error correction codes, $\mathsf{ECCEnc}, \mathsf{ECCDec}$, the conversion between a message $\boldsymbol{m} \in \{0,1\}^{l_m}$ and its encoding $\widehat{\boldsymbol{m}} \in \{0,1\}^{l_v}$ is provided, wherein l_v is a positive integer denoting the length of the encoding and depending on the specific choice of the parameter settings.

The algorithm $\mathsf{LAC.KG}$ randomly generates a pair of public key and secret key (pk, sk).

Algorithm 1 $\mathsf{LAC.KG}()$

```
Ensure: A pair of public key and secret key $(pk, sk)$.
$\mathsf{seed}_{\boldsymbol{a}} \xleftarrow{\$} \mathcal{S}$
$\boldsymbol{a} \leftarrow \mathsf{Samp}(U(R_q); \mathsf{seed}_{\boldsymbol{a}}) \in R_q$
$\boldsymbol{s} \xleftarrow{\$} \Psi_\sigma^{n,h}$
$\boldsymbol{e} \xleftarrow{\$} \Psi_\sigma^{n,h}$
$\boldsymbol{b} \leftarrow \boldsymbol{as} + \boldsymbol{e} \in R_q$
return $(pk := (\mathsf{seed}_{\boldsymbol{a}}, \boldsymbol{b}), sk := \boldsymbol{s})$
```

The algorithm $\mathsf{LAC.Enc}$ on input pk and a message $\boldsymbol{m}$, encrypts $\boldsymbol{m}$ with the randomness seed. In case that seed is not given, the process is randomized. Otherwise, the encryption is deterministic for the same seed. The subroutine ECCEnc converts the message $\boldsymbol{m}$ into a codeword $\widehat{\boldsymbol{m}}$.

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

```
Ensure: A ciphertext $\boldsymbol{c}$.
$\boldsymbol{a} \leftarrow \mathsf{Samp}(U(R_q); \mathsf{seed}_{\boldsymbol{a}}) \in R_q$
$\widehat{\boldsymbol{m}} \leftarrow \mathsf{ECCEnc}(\boldsymbol{m}) \in \{0,1\}^{l_v}$
$(\boldsymbol{r}, \boldsymbol{e}_1, \boldsymbol{e}_2) \leftarrow \mathsf{Samp}(\Psi_\sigma^{n,h}, \Psi_\sigma^{n,h}, \Psi_\sigma^{l_v}; \mathsf{seed})$
$\boldsymbol{c}_1 \leftarrow \boldsymbol{ar} + \boldsymbol{e}_1 \in R_q$
$\boldsymbol{c}_2 \leftarrow (\boldsymbol{br})_{l_v} + \boldsymbol{e}_2 + \lfloor \frac{q}{2} \rceil \cdot \widehat{\boldsymbol{m}} \in \mathbb{Z}_q^{l_v}$
return $\boldsymbol{c} := (\boldsymbol{c}_1, \boldsymbol{c}_2) \in R_q \times \mathbb{Z}_q^{l_v}$
```

The algorithm $\mathsf{LAC.Dec}$ 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. When there is a decryption error, the returned message $\boldsymbol{m} \notin \mathcal{M}$.

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

```
Ensure: A plaintext $\boldsymbol{m}$.
$\boldsymbol{u} \leftarrow \boldsymbol{c}_1 \boldsymbol{s} \in R_q$
$\widetilde{\boldsymbol{m}} \leftarrow \boldsymbol{c}_2 - (\boldsymbol{u})_{l_v} \in \mathbb{Z}_q^{l_v}$
for $i = 0$ to $l_v - 1$ do
    if $\frac{q}{4} \le \widetilde{\boldsymbol{m}}_i < \frac{3q}{4}$ 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 Formal Security

Following the result of [47], 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 [37,38] and quantum random oracle model [44]. 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.

4 Parameter Selection

Almost all lattice based key exchanges and public key encryptions, except for NTRU based ones, follow a similar framework from [32,56,18,11]. We have a set of theoretical results on the choice of rings, moduli, errors, etc [58,57,62] 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 [18,11,57,19,45,63], and a main differentiator in most NIST-PQC submissions [2]. 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 will work on reducing the modulus. Note that the modulus cannot be too small; it needs to be large enough to tolerant the errors during decryption. Prior to our work, 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 for choosing $q \equiv 1 \bmod 2n$ 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, $x^n + 1 \in \mathbb{Z}_{251}[x]$ 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 [15]. In that spirit, and also for the sake of simplicity, we choose the min-split modulus $q = 251$ for our scheme.

Remark 1. 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 [58] 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 [8,10], 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. For instance, very recently, Lyubashevsky et al. [] proposed new tricks that partially enable NTT for max-split rings. We leave the concrete security of those types of modulus to future research.

4.2 The Errors and Secrets Distribution

There are two rules for the choice of the distribution for the error and secret vector of the poly-LWE problem. Firstly, the errors and the secrets must be large enough to guarantee the hardness of the poly-LWE problem. Secondly, the errors and the secrets must be small enough to guarantee the correctness of the decryption algorithm. In literatures, there are mainly two families of distributions that satisfy the average/worst case reduction theorem [60,50], namely, discrete Gaussian distribution [50,12] and centered binomial distribution [19]. Gaussian distribution consumes lots of entropy, is hard to implement (in constant time), and is also vulnerable to memory based side channel attacks [22] when implemented with look-up tables [34]. Therefore, we opt to use the centered binomial distribution for our scheme.

In the implementation, as described in [12], 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. 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. 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. Then we can get the distribution Ψ_1. Besides, we also use a narrower distribution $\Psi_{\frac{1}{2}}$.

1. Ψ_1: $\Pr[x = 0] = 1/2$, $\Pr[x = \pm 1] = 1/4$.
2. $\Psi_{\frac{1}{2}}$: $\Pr[x = 0] = 3/4$, $\Pr[x = \pm 1] = 1/8$.

As pointed out by Alperin-Sheriff in the comments to LAC [2], when centered binomial distribution is used, the adversary can increase the decryption error rate by finding high hamming weight random vectors through pre-computation. The direct approach to resist this attack is to decrease the error rate by using a more powerful error correction code. However, correcting more errors will affect the efficiency of the error correction code.

To make LAC immune high hamming weight attack in a more efficient manner, we use n-ary centered binomial distribution $\Psi_{\sigma}^{n,h}$ with fixed Hamming weight h for the error and secret vectors $\boldsymbol{s}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{e}_1$. Since $\boldsymbol{e}_2$ only has slight influence on the decryption error rate, we still use the more efficient standard centered binomial distribution for $\boldsymbol{e}_2$. Concretely, we use the following distributions:

- LAC-128: Sample $\boldsymbol{s}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{e}_1$ from $\Psi_1^{n,h}$, $\boldsymbol{e}_2$ from Ψ_1^n, with $n = 512, h = 256$.
- LAC-192: Sample $\boldsymbol{s}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{e}_1$ from $\Psi_{\frac{1}{2}}^{n,h}$, $\boldsymbol{e}_2$ from $\Psi_{\frac{1}{2}}^n$, with $n = 1024, h = 256$.
- LAC-256: Sample $\boldsymbol{s}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{e}_1$ from $\Psi_1^{n,h}$, $\boldsymbol{e}_2$ from Ψ_1^n, with $n = 1024, h = 512$.

We remark that, when the hamming weight of the secret or error vector is fixed as the expectation of the standard centered binomial distribution, it only brings a very small effect to its entropy. Concretely, we compute the entropy of the distributions we used as follows:

- LAC-128: The entropy of $\boldsymbol{s}$ from Ψ_1^n with $n = 512$ is 768, when the hamming weight is fixed as $h = 256$ with 128 ones and 128 minus-ones, its entropy is about $\log\left(\binom{512}{128} \cdot \binom{384}{128}\right) \approx 758$.
- LAC-192: The entropy of $\boldsymbol{s}$ from $\Psi_{\frac{1}{2}}^n$ with $n = 1024$ is 1086, when the hamming weight is fixed as $h = 256$ with 128 ones and 128 minus-ones, its entropy is about $\log\left(\binom{1024}{128} \cdot \binom{896}{128}\right) \approx 1077$.
- LAC-256: The entropy of $\boldsymbol{s}$ from Ψ_1^n with $n = 1024$ is 1536, when the hamming weight is fixed as $h = 512$ with 256 ones and 256 minus-ones, its entropy is about $\log\left(\binom{1024}{256} \cdot \binom{768}{256}\right) \approx 1525$.

So using fixed weight centered binomial distribution will not affect the security reduction and concrete security evaluation of LAC. The implementation of fixed weight centered binomial distribution is also very simple. We only need to randomly set $h/2$ elements of the vector to be 1, $h/2$ elements to be -1 and others to be 0.

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:

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

Let $\boldsymbol{w} = (\boldsymbol{er} - \boldsymbol{e}_1\boldsymbol{s})_{l_v} + \boldsymbol{e}_2$, 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 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:

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

Suppose that 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 $\boldsymbol{m}$:

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

As pointed out by D'Anvers [30], when single bit error rate δ is too large, we can not assume that the coefficients of $\boldsymbol{w}$ are independent from each other. The theoretical dependence model and experiment results of D'Anvers show that the dependence mainly comes from the norm of $\boldsymbol{s}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{e}_1$. When fix hamming weight distribution is used for $\boldsymbol{s}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{e}_1$, their norms are also fixed and the main source of dependence is removed. So we can assume that the coefficients of $\boldsymbol{w}$ are independent from each other.

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 [12] 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.

4.5 Recommended Parameter Categories

We recommend the following parameter sets in Table 2, with respect to three categories of NIST post-quantum standardization project [2], namely, the equivalent security level of AES128, AES192 and AES256.

Categories	n	q	**dis**	**ecc**	l_m	**pk**	**sk**	**ct**	**bit-er**	**dec-er**
LAC-128	512	251	$\Psi_1^n, \Psi_1^{n,\frac{n}{2}}$	$[511, 256, 33]$	256	544	512	712	$2^{-12.61}$	2^{-116}
LAC-192	1024	251	$\Psi_{\frac{1}{2}}^n, \Psi_{\frac{1}{2}}^{n,\frac{n}{4}}$	$[511, 256, 17]$	256	1056	1024	1188	$2^{-22.27}$	2^{-143}
LAC-256	1024	251	$\Psi_1^n, \Psi_1^{n,\frac{n}{2}}$	[511,256,33]+D2	256	1056	1024	1424	$2^{-12.96}$	2^{-122}

dis	secret and noise distributions	**ecc**	error correction code
l_m	message length	**sk**	secret key size (bytes)
pk	public key size (bytes)	**ct**	ciphertext size (bytes)
bit-er	single bit error rate without BCH	**dec-er**	decryption error rate

Table 2. Recommended parameter of LACv2

According to the recommended parameter sets, the key size and ciphertext size of the CCA version of LAC are listed as below.

Categories	pk size(bytes)	sk size (bytes)	ciphertext size (bytes)
LAC-128	544	1056	712
LAC-192	1056	2080	1188
LAC-256	1056	2080	1424

Table 3. Key and ciphertext size of the CCA version of LACv2

Concretely, dimensions $n = 512$ and $n = 1024$ with basic error distributions Ψ_1^n and $\Psi_1^{n,n/2}$ discussed as above are for the low security level LAC-128 and the high security level LAC-256, respectively. To get the middle security level LAC-192, we use a smaller secret and noise distribution $\Psi_{\frac{1}{2}}^n, \Psi_{\frac{1}{2}}^{n,n/4}$ (and its) 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 [48], 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. We have $l_m = 256$ in our setting. Next, for $l_m = 256$, the minimum available BCH code length l_n is 511. Lastly, for the low security level LAC-128 and the high security level LAC-256, we choose $l_d = 33$ which allows us to correct 16 bits of errors at most. The redundant data due to this error correction code is 18 bytes. And for LAC-192, we use $l_d = 17$ which allows to correct up to 8 errors, and the redundancy is 9 bytes.

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 discarded. 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. One may simply store a 32 bytes seed for the secret key to minimize storage, at a cost of slightly slower decryption. 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 $2n+32$. 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 discarded). For LAC-128 parameter set, $l_v = l_m + 18 \times 8$, where 18 is the size of the redundant data. For LAC-192, $l_v = l_m + 9 \times 8$, where 9 is the size of the redundant data. And for LAC-256, $l_v = (l_m + 18 \times 8) \times 2$ 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 (e.g. BKZ with (quantum) sieving[3]) are well understood. Since our secrets and errors are sparse, we also evaluate the cost of hybrid attacks [43,66,41].

We also consider dedicated attacks that target specific designs of our scheme, namely the subfield attacks and the high Hamming weight attacks. Those at-

[3] There has been some debate on the accuracy of the formula to calculate the concrete cost of sieving [54]. These discussions are irrelevant to our parameters since a) they do not change that fact that the generic attacks remain the best attacks for LAC, and b) the LACv2 parameters are derived from the more conservative side of the debates.

tacks are reported as comments to the Round 1 version of LACv1 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 [10,64] for a survey of known techniques. It has been shown that lattice reduction attacks utilizing the BKZ algorithm [25] are more powerful than exhaustive search, combinational and algebraic algorithms. For simplicity, following the analysis of [11], we focus primly on two embedding attacks that are commonly referred to as primal attack and dual attack. We also evaluate the cost for hybrid attacks. We summarize the security estimates in Table 4.

Algorithm	**Primal Attack**			**Dual Attack**			**Hybird Attack**	
	C	**Q**	**B**	**C**	**Q**	**B**	**C**	**Q**
LAC-128	148	135	509	147	133	505	148	141
LAC-192	288	261	986	286	259	978	278	266
LAC-256	323	293	1105	320	290	1095	336	320

C: Classical complexity **Q:** Quantum complexity
B: Block Size

Table 4. 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 [11], 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 [11], 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, [11] 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.

Hybrid Attack. At a high level, the hybrid attack takes the following steps.

- Firstly, one interprets the lattice basis $\mathbf{B} \in \mathbb{Z}^{n,n}$ as a concatenation of two matrices $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{Z}^{\ell,n} \times \mathbb{Z}^{k,n}$, with $\ell > n/2$ and $\ell + k = n$. Note that $\mathcal{L}(B_1)$ spans a sublattice of dimension ℓ.
- Then, one performs lattice reductions over this sublattice, to an extend that it allows for solving bounded distance decoding problem effectively.
- Next, one guesses linear combinations of row vectors from $\mathbf{B}_2$. If one has guessed the correct combination as the unique shortest vector (or their rotations) in $\mathbf{B}$, this vector will be a close vector to $\mathcal{L}(\mathbf{B}_1)$.
- Lastly, one uses the reduced basis of $\mathcal{L}(B_1)$ to solve the bounded distance decoding problem.

It is easily to see that, to get the best performance, one usually assumes the cost of lattice reductions is on the same order of guessing. What remains to be estimated is the individual cost for each step. There are a few subtlety on estimating the cost of the above procedure. To be more conservative, we make the following assumptions/decisions.

- In the estimation, the cost of solving CVP/BDD is often neglected; a good basis yields a polynomial time algorithm (for example, Babai's algorithms), which may adds a few bits complexity in practice.
- We use BKZ with quantum core sieving/enumeration to estimate the cost of BKZ.
- We also assume that if basis is "good", and if we have successfully guessed the correct combination, then, the probability of solving the CVP is exactly 1.

- To be more conservative on the searching side, we estimate the entropy in the guessing phase, and taking its square root (in accounting for Grover's algorithm) as the cost of guessing. This results into a lower bound that isn't achievable through classical MITM attacks. This also assumes vector additions incur a cost of 1.

The details of the hybrid attack parameters can be found in Table 5.

Algorithm	**Classical**					**Quantum**				
	ℓ	Block size	BKZ	Search	Total	ℓ	Block size	BKZ	Search	Total
LAC-128	718	508	148	148	148	743	553	141	140	141
LAC-192	1141	984	278	278	278	1191	984	267	266	267
LAC-256	1151	1103	336	336	336	1367	1103	320	319	320

Table 5. Parametrizing the hybrid attacks

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 4. The corresponding security is then estimated for the obtained blocksize. Note that in [7], Albrecht *et al.* independently evaluated the security for all (Ring-)LWE candidates, and their estimation matches ours in this paper.

5.2 Dedicated Attacks

We stress again that the two attacks we are about to discuss do not outperform generic attacks. Both attacks do not work for LACv1 or LACv2. Nonetheless, having high Hamming weight in LACv1 may open doors to other cryptoanalysis, and for conservative purpose, we revised this parameter in LACv2.

Subfield Attacks. The idea of exploiting subfields is known to the lattice community for years [14,6,16,46], and to use this idea to analyze LAC was firstly proposed by Alperin-Sheriff [54] 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 high level 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 (while the number of iterations does not); 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 was able to perform lattice reduction over the dimension-halved lattices, he 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 $\boldsymbol{A_g}$ is 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 $\boldsymbol{A}\boldsymbol{z} = 0 \bmod q$, 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 [27], 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-128, $(253.26, 6.29)$ for LAC-192 and $(358.42, 6.86)$ for LAC-256[4].

It is easy to verify that, the length of $\boldsymbol{z}$ will be larger than the solution of $\boldsymbol{A}\boldsymbol{z} = 0 \bmod q$ 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.

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

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 if $(\boldsymbol{r}, \boldsymbol{e}_1)$ 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 (and correspondingly, pre-computations), an attacker may obtain sufficient number of samples with higher Hamming weight secrets and errors. This may leak information on the secret key (although no concrete attack exploiting this leakage has been proposed yet).

Analysis. It has been shown that chosen plaintext secure version of (Ring-)LWE based schemes suffer from an reaction attack [35]. To address this vulnerability, most schemes rely on Fujisaki-Okamoto transformation [37,38,44] 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 LACv1 [2], Alperin-Sheriff pointed out that, for the LAC-256 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

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

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, within the security model of NIST-PQC. Next, for samples with such high Hamming weights, the decryption error rate for each bit of $\widetilde{m}_i$ is expected at

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

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 2. We argue that, as also pointed by D'Anvers [2], 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 [35].

Counter measures for LACv1*.* An easy fix is to reduce the message size to 256. 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, the decoding efficiency decreases drastically when large $l_t = 100$ is used. To resolve this problem, for LAC-256 we use the D2 error correction code [18,11] 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 30 bits of errors.

The upper bound of the decryption error rate in the case of high Hamming weight attack is presented as follow. We give the upper bound of 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. We conclude 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	BCH	Decryption Error Rate
LAC-128	512+206	2^{-128}	$2^{-9.59}$	[511,256,61]	2^{-133}
LAC-192	512+333	2^{-192}	$2^{-14.75}$	[511,256,31]	2^{-142}
LAC-256	1024+416	2^{-256}	$2^{-9.77}$	[511,256,61]	2^{-138}

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 6. Decryption error rate of high Hamming weight attack

Counter measures for LACv2*.* In LACv2 we have switched to n-ary centered binomial distribution with fixed Hamming weight. This makes LAC completely immune from high Hamming weight attacks and their potential variants.

6 Implementations and Performance

We implement LAC on Intel x64 platform. 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. However, we are able to improve the computational efficiency with other techniques. In this section, we present some highlights of our customized implementation, including:

- a general optimized polynomial multiplication method (as per NIST-PQC's request);
- optimizations based on AVX2 instructions.

Generalized Optimization Techniques. Since polynomial multiplication is the most time consuming operation in the implementation of LAC, the optimization of the polynomial multiplication is our main focus. We achieve the goal by simplifying the following aspects:

- **Integer multiplication.** As a basic operation of polynomial multiplication, the integer multiplication in the implementation of LAC can be optimized as bit operations. 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 $a_i \times 1 = a_i\&0\text{xff}$ and $a_i \times 0 = a_i\&0\text{x}00$.
  Further more, we can pack 4 items into one `uint64_t` data type. Then, the polynomial multiplication becomes simply $\mathbf{as} = \sum_{s_i=1} a_i - \sum_{s_i=-1} a_i$, and can be evaluated in parallel.
- **Modular operation.** Another expensive part, the modular operation, can also be optimized for our byte-level modulus. With $q < 256$, it is possible to pack 8 coefficients into a single `uint64_t` unit, in theory. However, 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 Optimization. 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. These bring higher parallelism, and we can obtain approximately 30x acceleration with this optimization.

Error Correction Our BCH error correction code is an adaption of [33]. We also provide a less efficient but constant time BCH implementation in the use cases where side-channel attacks are a concern.

Performance Our platform is ubuntu 16.04 operation system running on the Intel Core-i7-4770S (Haswell) @ 3.10GHz, memory 7.6GB, with Turbo Boost and Hyperthreading disabled. In each case we provide two values to describe the performance of the algorithm the CPU cycles and the microseconds (μs). The data are presented in Table 7 and 8 for completeness.

Categories	Key generation		Encryption		Decryption		Decryption(Con-BCH)	
	Cycles	Time	Cycles	Time	Cycles	Time	Cycles	Time
LAC.CPA-128	124,915	40.28	194,118	67.24	81,187	26.28	122,355	39.47
LAC.CPA-192	335,083	106.20	438,204	144.63	292,243	93.80	309,896	93.80
LAC.CPA-256	382,627	124.23	636,997	204.80	302,890	95.18	338,993	108.25
LAC.CCA-128	122,691	39.67	209,201	65.71	280,125	88.07	323,221	102.70
LAC.CCA-192	333,649	105.63	445,696	145.48	731,472	235.42	759,871	244.49
LAC.CCA-256	377,123	123.59	643,024	208.71	916,835	297.01	934,385	304.82

Table 7. Performance of optimized LAC.CPA and LAC.CCA

Categories	Key generation		Encryption		Decryption		Decryption(Con-BCH)	
	Cycles	Time	Cycles	Time	Cycles	Time	Cycles	Time
LAC.CPA-128	61,242	19.98	80,173	25.91	25,004	7.83	64,238	20.77
LAC.CPA-192	120,528	38.87	130,286	42.34	63,266	26.41	134,289	39.95
LAC.CPA-256	136,313	54.23	191,543	63.14	72,326	30.56	112,654	48.99
LAC.CCA-128	59,584	19.59	89,055	28.86	103,229	39.26	140,221	45.57
LAC.CCA-192	119,246	36.94	137,653	65.14	224,249	71.52	320,135	77.32
LAC.CCA-256	135,780	53.95	207,938	87.88	343,335	84.21	359,209	97.60

Table 8. Performance of LAC.CPA and LAC.CCA with AVX2

7 Conclusion

Better instantiations and implementations of (Ring) LWE based PKE/KEX schemes have been a major topic for post-quantum cryptography community for the past years. Building on top of previous results, our work pushes the size limitation of post-quantum cryptography further. In parallel to rounding, modular lattices and other proposal, our idea of byte size modulus plus heavy error corrections seems to be one of the right direction to eventually obtain an RSA and ECC replacement in the quantum regime.

References

1. Lac source code, `https://github.com/luxianhui007/LAC`
2. Nist post-quantum cryptography project. `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions`
3. Nist post-quantum cryptography project round-2. `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions`

4. 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
5. 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
6. 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
7. 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
8. 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
9. 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
10. 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
11. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Newhope without reconciliation. IACR Cryptology ePrint Archive 2016, 1157 (2016), http://eprint.iacr.org/2016/1157
12. 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
13. 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
14. Bernstein, D.: A subfield-logarithm attack against ideal lattices (2014), https://blog.cr.yp.to/20140213-ideal.html
15. 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

16. 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 $l_{|\Delta K|}(\frac{1}{2})$ 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
17. 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
18. 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
19. 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)
20. 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
21. 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
22. 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
23. 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
24. 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
25. 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
26. Dai, W., Whyte, W., Zhang, Z.: Optimizing polynomial convolution for ntruencrypt. IACR Cryptology ePrint Archive 2018, 229 (2018), http://eprint.iacr.org/2018/229
27. Daniele Micciancio, O.R.: Lattice-based cryptography. Tech. rep., https://cims.nyu.edu/ regev/papers/pqc.pdf (2008)

28. D'Anvers, J., Guo, Q., Johansson, T., Nilsson, A., Vercauteren, F., Verbauwhede, I.: Decryption failure attacks on IND-CCA secure lattice-based schemes. In: Public-Key Cryptography - PKC 2019 - 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14-17, 2019, Proceedings, Part II. pp. 565–598 (2019), https://doi.org/10.1007/978-3-030-17259-6_19
29. D'Anvers, J.P., Tiepelt, M., Vercauteren, F., Verbauwhede, I.: Timing attacks on error correcting codes in post-quantum secure schemes. Cryptology ePrint Archive, Report 2019/292 (2019), https://eprint.iacr.org/2019/292
30. D'Anvers, J., Vercauteren, F., Verbauwhede, I.: The impact of error dependencies on ring/mod-lwe/lwr based schemes. IACR Cryptology ePrint Archive 2018, 1172 (2018), https://eprint.iacr.org/2018/1172
31. D'Anvers, J., Vercauteren, F., Verbauwhede, I.: On the impact of decryption failures on the security of LWE/LWR based schemes. IACR Cryptology ePrint Archive 2018, 1089 (2018), https://eprint.iacr.org/2018/1089
32. 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
33. Djelic, I.: Bch source code, https://github.com/jkent/python-bchlib/tree/master/bchlib
34. 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
35. 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
36. 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
37. 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
38. 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
39. 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
40. 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
41. Göpfert, F., van Vredendaal, C., Wunderer, T.: A hybrid lattice basis reduction and quantum search attack on LWE. In: Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands,

June 26-28, 2017, Proceedings. pp. 184–202 (2017), https://doi.org/10.1007/978-3-319-59879-6_11

42. 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
43. Howgrave-Graham, N.: A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In: Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings. pp. 150–169 (2007), https://doi.org/10.1007/978-3-540-74143-5_9
44. 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
45. Jin, Z., Zhao, Y.: Optimal key consensus in presence of noise. CoRR abs/1611.06150 (2016), http://arxiv.org/abs/1611.06150
46. 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
47. 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
48. Lu, X., Liu, Y., Jia, D., Xue, H., He, J., Zhang, Z.: Lac. Tech. rep., https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/LAC.zip (2017)
49. 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
50. 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
51. 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
52. Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient oneway 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
53. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Advances in Cryptology - EUROCRYPT 2012 - 31st Annual Inter-

national 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

54. NIST: NIST PQC FORUM: LAC, https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/official-comments/LAC-official-comment.pdf
55. 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
56. 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
57. 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
58. 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
59. 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
60. 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
61. 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
62. 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
63. 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
64. 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
65. 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

66. Wunderer, T.: Revisiting the hybrid attack: Improved analysis and refined security estimates. IACR Cryptology ePrint Archive 2016, 733 (2016), http://eprint.iacr.org/2016/733