An Efficient Post-Quantum PKE from RLWR with Simple Security Proof

Parhat Abla[1,2(✉)] and Mingsheng Wang[1]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, China
parhat@iie.ac.cn

[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. In this paper, we propose a public-key encryption scheme based on the Ring Learning With Rounding (RLWR) problem. Our scheme is seen as RLWR based variant of Saber (NIST PQC standardization round 3 candidate scheme). The design motivation is to overcome the very involved security proofs of LWR based public-key encryption schemes. To simplify the previous very involved security proofs, we introduce an intermediate problem which is at least as hard as RLWE problem. In contradiction to the previous LWR based schemes, our construction shares simple and intuitive security proof. We first present an IND-CPA public-key encryption scheme, and then apply a variant of the Fujisaki–Okamoto transforms to create a CCA- secure KEM. Our parameterization of the final KEM and the reference implementation shows that the performance of our scheme is comparable with the NIST PQC standardization round 3 candidates.

Keywords: RLWR · Lattice · Post-quantum · Encryption

1 Introduction

Since the ground breaking work of Shor [27], there has been a rising interest in post-quantum cryptography. Recently the interest of constructing post-quantum cryptographic primitives going up another level by the announcement of National Institute of Standards and Technology (NIST) that looking towards the standardization of post-quantum cryptography [1]. Among the submissions to the NIST, most of the constructions are based on the lattice problems. Furthermore, three [6,17,29] of the four round 3 finalists (public-key encryption and key-encapsulation algorithms) are lattice based primitives.

Since the work of Ajtai [2] showed the worst-case to average-case reduction on lattice problems, lattice based cryptography obtains much of interest. Yet the cryptographic schemes are inefficient until Regev [26] introduced the Learning With Errors (LWE) problem. Although the work of [26] presented a reduction

J. Garcia-Alfaro et al. (Eds.): SecureComm 2021, LNICST 399, pp. 407–423, 2021.
https://doi.org/10.1007/978-3-030-90022-9_22

from LWE to the shortest vectors problem on lattice, yet the reduction involves quantum algorithms. The later works [12,22] improved the reduction and consider the LWE problems over rings (RLWE) [21,24]. Let $R := \mathbb{Z}[x]/(x^n + 1)$ for some n of power of 2, and R_q be the quotient ring R/qR, then given the pair $(a, b) \in R_q \times R_q$ where a is uniform over R_q, the RLWE problem asks to distinguish if the ring element b is from a uniform distribution over R_q or $b = a \cdot s + e$ for some $s \in R_q$ and e sampled from some distribution over R_q. There are many constructions of public key encryption schemes based on the LWE problem or the RLWE problem [3,6,11,17,19,23,28].

The work of [9] considered a deterministic variant of LWE problem, called Learning With Rounding (LWR), in which the error term e is fixed by the ring elements a and s, namely $e := a \cdot s - \frac{q}{p}\lfloor \frac{p}{q} a \cdot s \rceil$[1]. One obvious intuitive advantage of LWR over LWE is that there is no need to sample the error term. The work [9] showed that if the ring modulus q is exponentially larger than the rounding modulus p, then the LWR problem can be reduced to the underlying LWE problem, and thus reduced to the hard problems on lattices. However the reduction remains valid for polynomially sized modulus q if there is a restriction on the number of LWR samples [4,5,10,20].

Contributions. Intuitively, the current RLWR based public key encryption schemes [7,17] use two-fold rounding operation in the encryption procedure. In other words, the public keys of [7,17] are generated by proceeding a rounding operation from R_q to R_p. Then the encryption algorithm proceeds a further rounding operation to the public keys (say from R_p to R_t) to disguise the message. From the hardness assumption of LWR problems, the outputs from the rounding operation should be uniformly distributed over the corresponding rings (R_p or R_t). Yet the security proof is more involved [17] or obscure [7]. Another disadvantage is that the modulus q could be much larger if the modulus t (corresponding to the second rounding) is not small, and this heavily affects the efficiency of the scheme.

In this work, we construct a public key encryption scheme based on the RLWR problem. The main design motivation of the scheme is that our scheme shares more simple security proof. As mentioned above, previous public key encryption schemes based on LWR problems use two-fold rounding. Yet in this paper, we overcome this by introducing an intermediate problem called LWR with auxiliary error. This intermediate problem is different from the RLWE problem, yet it is at least as hard as RLWE problem. We show that one rounding operation (R_q to R_p) is sufficient for the security proof of our construction, and the security proof is really simple. We also provide the IND-CCA secure key encapsulation and reference implementation of our scheme. We use the latest techniques [13] to accelerate the polynomial multiplications. The performance shows that our scheme has some advantages over the 3rd round NIST post-quantum cryptography standardization candidates.

[1] For a real $x \in \mathbb{R}$, $\lceil x \rfloor$ denotes the nearest integer to x. q and p are ring modulus such that $p < q$, mostly we require $p|q$.

Organization. In Sect. 2, we introduce preliminaries; we present our IND-CPA public key encryption scheme in Sect. 3, and provide asymptotic correctness as well as the security proof of our IND-CPA scheme. In Sect. 4, we present an IND-CCA secure key encapsulation scheme, and provide correctness and security results in the classic random oracle model and quantum oracle model, respectively. We present parameterizations and performances of our scheme in the last section.

2 Preliminaries

Notations. Let $\mathbb{R}$ be the set of real numbers, $\mathbb{Z}$ be the set of integers. For a real number $x \in \mathbb{R}$, use $\lfloor x \rfloor$ to denote the largest integer that $\leq x$, use $\lceil x \rfloor$ to denote the integer $\lfloor x+1/2 \rfloor$, and use $\lceil x \rceil$ to denote the smallest integer that $\geq x$. We use upper-case bold letters to denote matrices (e.g., $\vec{A}$). For a probability distribution χ, we use $x \leftarrow \chi$ to denote that x is sampled from distribution χ. For a set S, we use $x \xleftarrow{\$} S$ to denote that x is sampled from uniform distribution over the set S. For a polynomial ring R, we use $\|a\|_p$ to denote the p-norm of corresponding coefficient vector of $a \in R$, we omit the subscript p if $p = 2$. The function $\mathsf{negl}(\cdot)$ denotes the negligible function, that is $\mathsf{negl}(\lambda) < \frac{1}{\lambda^c}$ for the parameter λ and any constant c. We say an event happens overwhelmingly if the probability that the event not happens is negligible.

2.1 Cryptographic Definitions

Here we recall some definitions of public key encryption scheme that we will use in the following sections.

Public Key Encryption.

Definition 2.1. *A public key encryption scheme* Π_{PKE} *consist of algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *as follows:*

$\mathsf{KeyGen}(1^\lambda) \rightarrow (\mathsf{pk}, \mathsf{sk})$**:** *On input the security parameter, it outputs the public key* pk *and secret key* sk.

$\mathsf{Enc}(m, pk) \rightarrow \mathsf{ct}$**:** *On input the public key* pk *and the message* m*, it outputs the ciphertext* ct *corresponding to the message* m.

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \rightarrow m$**:** *On input the secret key* sk *and ciphertext* ct*, it outputs the decrypting message* m *corresponding to the ciphertext* ct.

Correctness. For a PKE scheme Π_{PKE}, we say it is $\delta-$correct if the following holds for the parameter $\delta \geq 0$:

$$\mathbf{E}\left[\max_{m \in \mathcal{M}} \mathsf{Pr}\left[m' \neq m \middle| \begin{array}{l} \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, m); \\ m' \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \end{array}\right]\right] \leq \delta,$$

where $\mathcal{M}$ is the message space, and the expectation is taken over the $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$.

Security. For any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ against a PKE scheme in the indistinguishability under chosen-plaintext attacks (IND-CPA) security game, denoted $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A})$, is defined as

$$\mathsf{Pr}\left[b' = b \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda); \\ (m_0, m_1) \leftarrow \mathcal{A}(pk); \\ b \leftarrow \{0,1\}; c^* = \mathsf{Enc}(\mathsf{pk}, m_b); \\ b' \leftarrow \mathcal{A}(c^*, pk) \end{array}\right] < \mathsf{negl}(\lambda).$$

Since the adversary can encrypt any message by itself using the public key, the IND-CPA security is a passive security notion that the adversary doesn't interact with the party who owns the secret key. A natural security model extension is the indistinguishability under chosen-ciphertext attacks (IND-CCA) security, in which the adversary is allowed to query the decryption algorithm except on the challenge ciphertext c^*. Note that there are many works on transforming a IND-CPA secure PKE to IND-CCA secure one by using classic random oracles [15] or quantum accessible oracles [16,18].

2.2 Lattices and Distributions

Lattices. An n-dimensional lattice is a discrete subgroup in the space $\mathbb{R}^n$. For a matrix $\vec{B} = [b_1, \cdots, b_n]$ of column vectors are linearly independent, a lattice generated by the matrix $\vec{B}$ is the integer combinations of the column vectors of $\vec{B}$.

Gaussians. For a real $s > 0$, we define the n-dimensional gaussian function with parameter s as $\rho_s(x) := \exp(-\pi \frac{\|\vec{x}\|^2}{s^2})$. For any n-dimensional vector $\vec{c}$, define the shifted gaussian by $\rho_{s,\vec{c}} := \exp(-\pi \frac{\|\vec{x}-\vec{c}\|^2}{s^2})$. The distribution function of the spherical continuous gaussian D_s over $\mathbb{R}^n$ is proportional to ρ_s. For $\delta > 0$, we call a random variable X over $\mathbb{R}$ is δ-subgaussian with parameter $s > 0$, if for all $t \in \mathbb{R}$, the (scaled) moment-generating function satisfies: $E[e^{2\pi tX}] \leq e^{\delta} \cdot e^{\pi s^2 t^2}$. B-bounded symmetric random variable X(i.e., $|X| \leq B$) is 0-subgaussian with parameter $B\sqrt{2\pi}$, and thus the centered binomial distribution is also subgaussian. Therefore in this paper we focus on binomial distribution as it is close to small gaussian distributions [8].

Rings. Let the ring $R := \mathbb{Z}[x]/\Phi(x)$ for some cyclotomic polynomial $\Phi(x)$, and let R_q be the quotient ring R/qR for some modulus q, we say a is sampled from R (or R_q) by binomial distribution with parameter η, we mean by that the corresponding coefficient vector of a is sampled from the n-dimensional binomial distribution over integers, simply write $a \leftarrow \mathsf{Bin}_\eta$. In this paper, we set $\Phi(x) := (x^n + 1)$ for some n of a power of 2. For a set $S \in \mathbb{Z}$, we use R_S to denote the set of ring elements such that each coefficients are in the set S, and we let $x \leftarrow \mathcal{U}(S)$ to denote the uniform distribution over the set of ring elements R_S. We use $\|a\|_p$ to denote the p-norm of the coefficient vector the ring element $a \in R$. Similarly, $\|a\|_\infty$ denotes the largest coefficient of the ring element a.

2.3 LWR Problem

Here in this section, we recall the ring version of the RLWR problem. Let $\lceil * \rfloor_{q \to p}$ be the rounding function that for an integer $x \in \mathbb{Z}$ $\lceil x \rfloor_{q \to p} = \lceil \frac{q}{p} x \rfloor$, and it applies over the ring elements coefficient-wise. We recall the RLWE distribution over the ring R_q as follow.

Definition 2.2. *The* $R\text{-}\mathsf{LWR}_{n,p,q,\chi}$ *distribution is the distribution of the pair* $(a, b) \in R_q \times R_p$*, where* a *is uniform over* R_q *and* $b = \lceil a \cdot s \rfloor_{q \to p}$ *for some* $s \leftarrow \chi$*.*

We recall the $R\text{-}\mathsf{LWR}_{n,p,q,\chi}$ problem as follows.

Definition 2.3. *Given a pair* $(a, b) \in R_q \times R_p$ *for the random* $a \leftarrow R_q$*, the* $R\text{-}\mathsf{LWR}_{n,p,q,\chi}$ *problem asks to distinguish if the pair is from the* $R\text{-}\mathsf{LWR}_{n,p,q,\chi}$ *distribution or the uniform distribution over* $R_q \times R_p$*.*

The LWR (RLWR) problem was introduced in [9]. However, the work of [9] showed the hardness of RLWR problem for the sub-exponential modulus q. Note that the work of [9] considered the statistical property in the reduction, thus the hardness of LWR is regardless of the number of the LWR samples. Later works [4,5,10,20] showed that polynomial modulus is enough to show the hardness of LWR problem if we consider the polynomial number of samples. Note that polynomial number of LWR samples are sufficient for the encryption schemes and other cryptographic protocols, thus a polynomially modulus is sufficient for this work.

3 The Basic Encryption Scheme

In this section, we present detailed description of our basic encryption scheme. In addition, we further provide the asymptotic correctness and the IND-CPA security of the scheme.

3.1 The Scheme Description

The description of our encryption scheme is shown as in the Algorithm 1, 2, and 3. The set param contains all the scheme related parameters such as dimension of the ring, modulus p, q, message space, security parameter λ, and others. On input a λ-bit length seed, the deterministic algorithm $\mathsf{Gen}_a(\cdot)$ outputs a random ring element $a \in R_q$. On input a random seed, the algorithm Bin_η outputs a ring element whose each coefficients are sampled from binomial distribution of parameter η. On input an integer $x \in \mathbb{Z}_q$, the rounding function $\lceil x \rfloor_{q \to p}$ is defined as $\lceil x \rfloor_{q \to p} := \lceil \frac{p}{q} x \rfloor \mod p$. If the input is a ring element, the rounding function applies coefficient-wise. Note that we want the modulus p and q satisfies $2p|q$, thus the quantity $\frac{q}{2p}$ is an integer.

Algorithm 1: IND-CPA.KeyGen(param)

$(seeds) \xleftarrow{\$} \{0,1\}^{\lambda}$
$a \leftarrow R_q$
$s \leftarrow \mathsf{Bin}_{\eta}(seeds)$
$b := \lceil a \cdot s \rfloor_{q \to p}$
Return: $\mathsf{pk} := (a, b), \mathsf{sk} := s$

Algorithm 2: IND-CPA.Enc($\mathsf{pk} = (a, b), \mu \in R_2$)

$seedr \xleftarrow{\$} \{0,1\}^{\lambda}$
$r \leftarrow \mathsf{Bin}_{\eta}(seedr)$
$u \xleftarrow{\$} \mathcal{U}\left(\left[\frac{-q}{2p}, \frac{q}{2p}\right) \cap \mathbb{Z}\right)$
$c_0 := \lceil a \cdot r \rfloor_{q \to p}$
$c_1' := \frac{q}{p} b + (u \mod q/p) \mod q.$
$c_1 := \lceil c_1' \cdot r \rfloor_{q \to p} + \lceil \frac{p}{2} \rfloor \mu$
Return: $\mathsf{ct} := (c_0, c_1)$

Algorithm 3: IND-CPA.Dec($\mathsf{ct} = (c_0, c_1), \mathsf{sk} = s$)

$\mu' := c_1 - c_0 \cdot s$
$\mu'' := \lceil \mu' \rfloor_{p \to 2}$
Return: μ''

For simplicity we didn't explicit the param in the input of the encryption and the decryption algorithms, yet they implicitly contain it.

Remark 3.1. *Note that the message space in the encryption algorithm is R_2, However it can be enlarged to R_t for some small t by the cost of increasing the decryption failure probability. Compressing the ciphertexts also applicable to above encryption scheme, however it increases the failure probability as well.*

3.2 Correctness and Security

Correctness. The correctness of above scheme described in the Algorithm 1 to Algorithm 3 is given by the following theorem. In Sect. 5, we will show the correctness of the scheme in the concrete parameter settings.

Theorem 3.2. *Let the rounding modulus $p \geq 10$ Let s, r, u be the random variables as in the Algorithm 1 and Algorithm 2, u is uniform over $R_{[\frac{-q}{2p}, \frac{q}{2p})}$, and $err_b \leftarrow \chi_s$, $err_{c_0} \leftarrow \chi_r$, $err_{c_1} \leftarrow \chi_{s,r,u}$, where the distributions are defined below. Let δ be the probability that*

$$\delta := \mathsf{Pr}[||(err_b + u) \cdot r + err_{c_1} - err_{c_0} \cdot s||_{\infty} \geq \frac{q}{5}], \tag{1}$$

then the scheme described in the Algorithm 1 to Algorithm 3 is $(1-\delta)$*-correct. For a random ring element* $a \in R_q$*, the distribution* χ_x *indexed by* $x \in R$ *is the distribution of* $\left(a \cdot x - \frac{q}{p}\lceil a \cdot x \rfloor_{q\to p}\right)$*, and* $\chi_{x,y,z}$ *indexed by* $x, y \in R$ *is the distribution of* $\left((\frac{q}{p}\lceil a \cdot x \rfloor_{q\to p} + z)y - \frac{q}{p}\lceil(\frac{q}{p}\lceil a \cdot x \rfloor_{q\to p} + z)y \rfloor\right)$.

Proof. From the description of the Algorithm 3, it's not hard to see that the scheme is $(1-\delta)$-correct if $\mathsf{Pr}[||\mu'' - \mu||_\infty > 0] \leq \delta$. Thus showing $\mathsf{Pr}[||\mu'' - \mu||_\infty \leq 0] \leq 1 - \delta$ is sufficient to the theorem.

The description of Algorithm 1 and Algorithm 2 tell that:

$$\frac{q}{p}b = a \cdot s + err_b, \quad c_0 = \frac{p}{q}(a \cdot r + err_{c_0}), \quad c_1 = \frac{p}{q}(c_1' \cdot r + err_{c_1}) + \lceil \frac{p}{2} \rceil \mu,$$

for some $err_b, err_{c_0}, err_{c_1} \in R_q$, and are distributed accordingly to the distribution χ_s, χ_r and $\chi_{s,r}$ as the theorem assumption. From the decryption procedure, we have following

$$\begin{aligned} \mu' &= \tfrac{p}{q}\left(c_1' \cdot r + err_{c_1}\right) + \lceil \tfrac{p}{2} \rceil \mu - \tfrac{p}{q}(a \cdot r + err_{c_0}) \cdot s && (2)\\ &= \tfrac{p}{q}\left((\tfrac{q}{p}b + u) \cdot r + err_{c_1}\right) - \tfrac{p}{q}(a \cdot r \cdot s + err_{c_0} \cdot s) + \lceil \tfrac{p}{2} \rceil \mu && (3)\\ &= \tfrac{p}{q}\left((err_b + u) \cdot r + err_{c_1} - err_{c_0} \cdot s\right) + \lceil \tfrac{p}{2} \rceil \mu, && (4) \end{aligned}$$

where the first equality is by the decryption algorithm that $\mu' = c_1 - c_0 \cdot s$; the second equality is by the definition of $c' = \frac{q}{p}b + u$ in Algorithm 2; the last equality is by $\frac{q}{p}b = a \cdot s + err_b$ and rearranging them. We further have that

$$\mu'' = \lceil \mu' \rfloor_{p\to 2} = \lceil \frac{2}{q}\left((err_b + u) \cdot r + err_{c_1} - err_{c_0} \cdot s\right) + \frac{2}{p}\lceil \frac{p}{2} \rceil \mu \rfloor. \quad (5)$$

Since $||(err_b + u) \cdot r + err_{c_1} - err_{c_0} \cdot s||_\infty \geq \frac{q}{5}$ holds with probability δ and $p \geq 10$, hence $||\mu'' - \mu||_\infty \leq \lceil \frac{2}{5} + \frac{1}{p} \rfloor = 0$ holds with probability $1 - \delta$. This completes the proof. □

Security. The security of the encryption scheme given in Algorithm 1 to Algorithm 3 can be reduced to the hardness of RLWR problem. To show the security of the scheme, we introduce an intermediate problem RLWR with auxiliary error (RLWE-AE) problem. The definition of RLWE-AE problem is as follow.

Definition 3.3. *For integers* p, q *such that* $2p|q$ *and a distribution* Bin *over* R_q*, the* R-$\mathsf{LWR\text{-}AE}_{n,p,q,\chi}$ *problem gives as challenge* $(a, b) \in R_p \times R_p$*, where* $a \xleftarrow{\$} R_p$*, and asks to decide if* $b = \lceil (\frac{q}{p}a + u) \cdot s \rfloor_{q\to p}$ *for some* $s \leftarrow \mathsf{Bin}$ *and* $u \xleftarrow{\$} \mathcal{U}([\frac{-q}{2p}, \frac{q}{2p}) \cap \mathbb{Z})$*, or* b *is from uniform distribution over* R_p*.*

One obvious difference between this problem and the RLWE problem is that the errors here are uniformly distributed while the errors in the RLWE problems are from the gaussian distribution. In practice sampling a uniform element is easier than sample a gaussian element. The following lemma shows the hardness of R-$\mathsf{LWR\text{-}AE}$ problem. The proof of above lemma given in the Appendix A.1

Lemma 3.4. *Let p, q be the integers such that $2p|q$, and χ be some distribution, then the RLWR with auxiliary error problem R-LWR-$\mathsf{AE}_{n,p,q,\chi}$ is no easier than the RLWR problem R-$\mathsf{LWR}_{n,p,q,\chi}$.*

The security of our encryption scheme, described in the Algorithm 1, 2 and 3, is given by the following theorem.

Theorem 3.5. *The encryption scheme given in the Algorithm 1, 2 and 3 is IND-CPA secure if the underlying R-$\mathsf{LWR}_{2n,p,q,\mathsf{Bin}_\beta}$ problem is hard.*

Proof. We show the theorem by showing that if the underlying LWR problem is hard, then the ciphertexts (c_0, c_1) are computationally indistinguishable from the uniform elements in $R_p \times R_p$ regardless of the message encrypted, and thus the IND-CPA security is follows. To show this, we introduce following hybrid games.

Game_1	Game_2	Game_3
$seeds, seedr \xleftarrow{\$} \{0,1\}^{256}$	$seeds, seedr \xleftarrow{\$} \{0,1\}^{256}$	$seeds, seedr \xleftarrow{\$} \{0,1\}^{256}$
$a \xleftarrow{\$} R_q$	$a \xleftarrow{\$} R_q$	$a \xleftarrow{\$} R_q$
$s \leftarrow \mathsf{Bin}_\eta(seeds)$	$s \leftarrow \mathsf{Bin}_\eta(seeds)$	$s \leftarrow \mathsf{Bin}_\eta(seeds)$
$b = \lceil a \cdot s \rfloor_{q \to p}$	$b \xleftarrow{\$} R_p$	$b \xleftarrow{\$} R_p$
$r \leftarrow \mathsf{Bin}_\eta(seedr)$	$r \leftarrow \mathsf{Bin}_\eta(seedr)$	$r \leftarrow \mathsf{Bin}_\eta(seedr)$
$c_0 = \lceil a \cdot r \rfloor_{q \to p}$	$c_0 = \lceil a \cdot r \rfloor_{q \to p}$	$c_0 \xleftarrow{\$} R_p$
$u \leftarrow \mathcal{U}\left[\frac{-q}{2p}, \frac{q}{2p}\right)$	$u \leftarrow \mathcal{U}\left[\frac{-q}{2p}, \frac{q}{2p}\right)$	$u \leftarrow \mathcal{U}\left[\frac{-q}{2p}, \frac{q}{2p}\right)$
$c_1' := \frac{q}{p} b + (u \bmod \frac{q}{p})$	$c_1' := \frac{q}{p} b + (u \bmod \frac{q}{p})$	$c_1' := \frac{q}{p} b + (u \bmod \frac{q}{p})$
$c_1 := \lceil c_1' \cdot r \rfloor_{q \to p} + \lceil \frac{p}{2} \rceil \mu$	$c_1 := \lceil c_1' \cdot r \rfloor_{q \to p} + \lceil \frac{p}{2} \rceil \mu$	$c_1 \xleftarrow{\$} R_p$
$\beta \leftarrow \{0,1\}$	$\beta \leftarrow \{0,1\}$	$\beta \leftarrow \{0,1\}$
if $\beta = 1$ **then**	**if** $\beta = 1$ **then**	**if** $\beta = 1$ **then**
return (a, b, c_0, c_1)	return (a, b, c_0, c_1)	return (a, b, c_0, c_1)
else $\beta = 0$	**else** $\beta = 0$	**else** $\beta = 0$
return $(a, b, \mathcal{U}(R_p \times R_p))$	return $(a, b, \mathcal{U}(R_p \times R_p))$	return $(a, b, \mathcal{U}(R_p \times R_p))$

Note that the Game_1 is subtle different from the original IND-CPA game that here in Game_1, the adversary $\mathcal{A}$ is given the public keys (a, b) and the challenge pair (c_0, c_1), and $\mathcal{A}$'s goal is to guess the random number β. We call $\mathcal{A}$ wins in Game_i if it correctly guesses the random bit β. The IND-CPA security of the scheme is obvious if $\mathcal{A}$'s advantage in Game_1 is negligible. Let $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_i}$ to denote the advantage of $\mathcal{A}$ in the Game_i for $i \in \{1, 2, 3\}$, namely $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_i} = |\mathsf{Pr}[\mathcal{A} \text{ win } \mathsf{Game}_i] - \frac{1}{2}|$. We also use $\mathsf{Game}_i \overset{c}{\approx} \mathsf{Game}_j$ to denote the indistinguishability of two games. To show the theorem, we have following lemmas and proofs are defer to Appendix A.3 and Appendix A.2.

Lemma 3.6. *If the R-$\mathsf{LWR}_{n,p,q,\mathsf{Bin}_\beta}$ problem is hard, then $\mathsf{Game}_1 \overset{c}{\approx} \mathsf{Game}_2$.*

Lemma 3.7. *If the R-$\mathsf{LWR}_{2n,p,q,\mathsf{Bin}_\beta}$ problem is hard, then $\mathsf{Game}_2 \overset{c}{\approx} \mathsf{Game}_3$.*

Above two lemmas show that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_1} \leq \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2} + \mathsf{negl}(n) \leq \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_3} + \mathsf{negl}(n) \leq \mathsf{negl}(n),$$

where the first and second inequalities are from Lemma 3.6 and Lemma 3.7; the last inequality is from the fact that $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_3} = 0$, this is because the 4-tuples (a, b, c_0, c_2) are uniform over $R_q \times R_p^3$ and independent of the random bit β. This completes the proof. □

4 The CCA Secure Scheme

In this section, we present our IND-CCA secure KEM construction, and we show its correctness and security results.

Scheme Description. Let $G : \{0,1\}^* \to \{0,1\}^{256\times 2}$ and $H : \{0,1\}^* \to \{0,1\}^{256}$ be the two hash functions, our KEM construction consists of 3 algorithms $(\mathsf{KeyGen}, \mathsf{Encaps}, \mathsf{Decaps})$ presented in the following Algorithm 4, 5 and 6. The key generation algorithm is the same as the Algorithm 1 except that here the secret key contains an extra 256 bit random string. The encapsulation and decapsulation algorithms are obtained by using a KEM variant of F-O transform [15,16] to our basic IND-CPA scheme given in the previous section.

Algorithm 4: $\mathsf{IND\text{-}CCA.KeyGen}(1^\lambda)$

1: $(seeds, z) \xleftarrow{\$} \{0,1\}^{256}$
2: $a \xleftarrow{\$} R_q$
3: $s \leftarrow \mathsf{Bin}_\eta(seeds)$
4: $b := \lceil a \cdot s \rfloor_{q\to p}$
5: **Return**: $\mathsf{pk} := (a, b), \mathsf{sk} := (s, z)$

Algorithm 5: $\mathsf{IND\text{-}CCA.Encaps}\,(\mathsf{pk} = (a, b),)$

1: $\mu \xleftarrow{\$} \{0,1\}^{256}$
2: $(\hat{K}, rand) := G(\mathsf{pk}|\mu)$
3: $(c_0, c_1) := \mathsf{IND\text{-}CPA.Enc}(\mathsf{pk}, \mathsf{Encode}(\mu); rand)$
4: $\mathsf{ct} := (c_0, c_1)$
5: $K := H(\hat{K}|\mathsf{ct})$
6: **Return**: (ct, K)

Algorithm 6: $\mathsf{IND\text{-}CCA.Decaps}\,(\mathsf{ct} = (c_0, c_1), \mathsf{pk} = (a, b), \mathsf{sk} = (s, z))$

1: $\mu' := \mathsf{Decode}(\mathsf{IND\text{-}CPA.Dec}(\mathsf{ct}, s))$
2: $(\hat{K}', rand') := G(\mathsf{pk}|\mu')$
3: $(c_0', c_1') := \mathsf{IND\text{-}CPA.Enc}(\mathsf{pk}, \mathsf{Encode}(\mu'); rand')$
4: $\mathsf{ct}' := (c_0', c_1')$
5: **if** $\mathsf{ct} = \mathsf{ct}'$ **then**
6: **Return**: $K := H(\hat{K}'|\mathsf{ct})$
7: **else**
8: **Return**: $K := H(z|\mathsf{ct})$

Correctness. Note that the randomness in the encryption procedure in Algorithm 5 is determined by the message to be encrypted, and thus the decryption failure probability is affected by the number of queries, denoted q_G, to the RO G. If the underlying IND-CPA encryption scheme is δ-correct, then the resulting scheme in the Algorithm 4, 5, and 6 is $q_G\delta$-correct [16].

Security. Note that using the variant of F-O transform [16], the final scheme can be proven IND-CCA secure in the random oracle model. The IND-CCA security of the encryption scheme described in the Algorithm 4, 5, and 6 is given by the following results.

Theorem 4.1 (ROM, [16]). *If the encryption scheme* PKE *described in the Algorithm 1, 2, and 3 is δ-correct, for any* IND-CCA *adversary $\mathcal{A}$ against the encryption scheme given in Algorithm 4, 5, and 6, and let q_G, q_H be the number of (might be quantum) queries made by $\mathcal{A}$ to the random oracles G and H, then there is an adversary $\mathcal{B}$ against the* PKE *such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathcal{A}} \leq \frac{2q_G + q_H + 1}{\sqrt{|\mathcal{M}|}} q_G \sqrt{\delta} + 3\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}), \tag{6}$$

where $\mathcal{M}$ is the message space.

Note that the above theorem is illustrates the IND-CCA security of the scheme assuming the adversary enables to query the RO's as much as it's desire. Yet the RO's are publicly available, consider the case where the adversaries may capable of quantum computers, we want the encryption scheme provides security in this scenario. Fortunately, the scheme in the Algorithm 4, 5, and 6 is designed by applying the paradigm of variants of F-O transform [15,16,18], and thus provide the IND-CCA security in the presence of quantum accessible ROs. The following theorem and the corollary describe this kind of security.

Theorem 4.2 (QROM, [18]). *If the encryption scheme* PKE *described in the Algorithm 1, 2, and 3 is δ-correct, for any* IND-CCA *adversary $\mathcal{A}$ against the encryption scheme given in Algorithm 4, 5, and 6, able to query the random oracles with quantum states, let q_G, q_H be the number of (might be quantum) queries made by $\mathcal{A}$ to the random oracles G and H, then there is an adversary $\mathcal{B}$ against the* PKE *such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathcal{A}} \leq 2q_H \frac{1}{\sqrt{|\mathcal{M}|}} + 4q_G\sqrt{\delta} + 2(q_G + q_H)\sqrt{\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B})}, \tag{7}$$

where $\mathcal{M}$ is the message space.

Note that the message space of our scheme is exponentially in security parameter and the basic encryption scheme described in the Algorithm 1, 2, and 3 is IND-CPA secure by Theorem 3.5, then the first and third term in the right hand side of the above inequation is negligible. Thus the encryption scheme described in Algorithm 4, 5, and 6 is IND-CCA secure if the encryption scheme PKE described in the Algorithm 1, 2, and 3 is overwhelmingly correct.

5 Parameter Settings and Implementation

In this section, we provide concrete parameters for our encryption scheme and performance of C reference implementation.

5.1 Parameter Settings

To instantiate the encryption scheme, we select the concrete parameters such that the resulted scheme has low decryption failure and provides desired bit security. However, setting the parameters according to the theoretical results definitely affects the performance of the scheme. Thus in practice we pay more attention to the concrete security of the scheme.

Decryption Failure. The asymptotic correctness result for the scheme is presented in the Theorem 3.2, yet in practice we need more precise estimation of the failure probability as the encryption scheme is instantiated with concrete parameters. For our encryption scheme in the concrete parameters, we first compute the distribution function of each error terms involved in the Theorem 3.2. Intuitively, the larger the error terms the harder to solve the corresponding LWR problem. However, increasing the errors result in increasing the decryption failure probability. Note that each coefficient of the error polynomial is discrete and upper bounded by small positive, thus we can compute the probability distribution of each coefficients of error terms. then we compute the overall distribution of the final error. Note that this procedure is independent of the running time of the scheme, and thus any tools can be used to compute this failure probability. Here we use a python script similar as [6,17] to estimate the failure probability of our encryption scheme in the concrete settings, and results are shown in Table 1.

Concrete Security. As we showed in the previous sections, our scheme is based on the ring LWR problem. However the best way to solve RLWR problem now is to treat it as a LWE problem with deterministic errors, and further estimate the security of corresponding LWE problem with same dimension and modulus. There are two types of attacks to estimate the concrete security of LWE based crypto-systems: primal attack and dual attack [25]. Apply this two types of attacks to estimate the concrete bit security of a crypto-system is the main stream in the lattice based literatures [6,17,25]. Recently, Dachman-Soled et al. [14] formalized the primal attack and found a novel improvement to attack a few NIST second round post-quantum cryptography standardization candidate schemes [7,29]. Thus we use both the folklore estimation method and the recent method to estimate the concrete bit security of our scheme. The bit security of our scheme in the concrete parameter settings is shown in the Table 1.

Parameterization. As we showed in the security proof of our scheme, we need the ring modulus q and rounding modulus p to satisfy $2p|q$. Thus we set the ring modulus q be a multiple of $2p$, e.g., $q = 2^k \cdot p$ for some integer $k \in \mathbb{Z}^+$. Here we choose them a power of 2, yet any other settings are possible. Concrete parameterization of our scheme is shown in the following Table 1, where the

$\log(q)$ or $\log(p)$ mean that logarithmic function of the modulus q or p. The η denotes the binomial parameter. If the classic computer used to solve the corresponding SVP problem respect to the LWR problem, we call it the classic security of the LWR problem (as showed in the Table 1), otherwise the quantum algorithms are used, then we call it the quantum security of the LWR problem as we split the security column into two sub-columns). The space complexity of the public keys and ciphertexts are computed in bytes (B). As shown in the Table 1, the Scheme1 reaches 128-bit classic security, and the Scheme2 reaches 256-bit classic security.

Table 1. Parameter settings of our schemes.

	n	$\log(q)$	$\log(p)$	η	Failure	Security	
						Classical	Quantum
Scheme1	512	11	8	2	2^{-115}	134	121
Scheme2	1024	13	10	4	2^{-128}	274	250

The following Table 2 shows the space complexity of our Schemes and NIST PQC standardization 3-rd round lattice based PKE finalist schemes. We compare them in 128-bit security settings and 256-bit security settings.

Table 2. Space complexity comparison with NIST PQC standardization lattice based PKE finalists.

	$\mathsf{pk}(B)$	$\mathsf{sk}(B)$	$\mathsf{ct}(B)$	Failure	Classic
Scheme1	608	768	1024	2^{-115}	134
LightSaber [17]	672	832	736	2^{-120}	118
Kyber512 [6]	800	1632	768	2^{-139}	118
ntruhps2048509 [29]	699	903	699	$2^{-214.3}$	118
Scheme2	1312	1888	2560	2^{-128}	274
FireSaber [6]	1312	1760	1472	2^{-168}	260
Kyber1024 [6]	1184	2400	1084	2^{-174}	256
ntruhrss701 [29]	1138	1418	1138	$2^{-213.9}$	118
ntruhps4096821 [29]	1230	1588	1230	2^{-769}	118

5.2 Implementation

Sampling. Our encryption scheme need two type of sampling: sampling from binomial distribution with small parameter, and sampling from uniform distribution over a small centered symmetric range. The binomial sampling procedure can be accomplished by sampling uniform bit strings. Namely, a sample

from binomial distribution Bin_η is sampled as follow: first sample a bit string $(str1_i, str2_i)_{i=1}^{\eta} \in \{0,1\}^\eta$, then output $\sum_{i\in[\eta]}(str1_i - str2_i)$. Since the modulus q and p are power of 2, the uniform sample procedure in the Algorithm 2 is by simply sample a $\log(\frac{q}{p})$ bit string, then subtract $\frac{q}{2p}$ form the integer representation of the bit string and output the result. Thus the whole sampling procedure is as quick as generating random strings. To sample a random ring element $a \in R_q$, we first feed *Output eXtendable Function* (OXF) a random seed to obtain large random bits, and then generate the random element a with these bits. Note that this method reduces randomness of a to the security of OXF, yet the public key size of our scheme is improved significantly.

Rounding. Since the scheme modulus q and p are set to be the power of 2, thus the rounding procedure as simple as bit shift operation.

Multiplication. As the settings of the modulus p and q be some power of 2 ease the sampling and rounding procedure, yet the multiplication operation is important as the those operation, and might even have more impact on the performance of the scheme. However, as our parameter settings, the polynomial multiplication can't be accelerated by the NTT. Yet we use Karatsuba algorithm to implement the polynomial multiplication, and the Table 3 shows the performance of our schemes.

Performance. All the cycle counts in Table 3 were obtained on Macbook pro equipped with Quad-Core Intel(R) Core(TM) i5-8257U CPU@3.9 Ghz (turbo boost on) processor. We use g++11 with optimization falgs *-march=native -O3 -fomit-frame-pointer -fwrapv -Qunused-arguments*. For the random oracles we use SHA3 functions (implemented in fips202). The performance is shown in the following Table 3.

Table 3. Performance comparison with NIST PQC standardization lattice based PKE finalists (in cycles).

	KeyGen	Enc	Dec
LightSaber [17]	39561	40154	32064
Kyber512 [6]	54189	57557	61409
ntruhps2048509 [29]	5695661	397765	576719
Scheme1	27741	54586	64617
FireSaber [6]	81327	84558	85517
Kyber512 [6]	123693	125461	137841
ntruhps4096821 [29]	15263307	770705	1434187
Scheme2	87092	180290	253419

The above table shows that the performance of our scheme is comparable with LightSaber and Kyber512. Note that the recent work of [13] showed that the Saber encapsulation can be improved about 22%, thus our schemes also

can be improved significantly by using the same techniques from [13]. We only provide reference C implementation, thus further optimizations are possible. Furthermore, we didn't consider the ciphertext compression, and thus a carefully designed integration of ciphertext compression with efficient implementation will result in more efficient scheme than ours. Note that the plain LWR version of our encryption scheme is similar to ours, thus we only consider the ring LWR version here.

Acknowledgement. Mingsheng Wang is supported by the Shandong Provincial Key Research and Development Program under Grant Number 2019JZZY020127.

A Appendix

A.1 Proof of Lemma 3.4

Proof. We show the lemma by contradiction. Namely, we show that if there is an algorithm $\mathcal{A}$ that can solve the problem $R\text{-}\mathsf{LWR}\text{-}\mathsf{AE}_{n,p,q,\chi}$, then we can construct a simulator Sim which solves the RLWR challenge with the same advantage. Our construction of Sim is as follows:

$\mathsf{Sim}(a, b)$

On input the pair $(a, b) \in R_q \times R_p$ of RLWR challenge, it gives the pair $(\lceil a \rfloor_{q\to p}, b)$ to the algorithm $\mathcal{A}$. If $\mathcal{A}$ outputs 1 (mean that the input pair $(\lceil a \rfloor_{q\to p}, b)$ is from the $R\text{-}\mathsf{LWR}\text{-}\mathsf{AE}_{n,p,q,\chi}$ distribution), then Sim also outputs 1 to mean that (a, b) is from the $R\text{-}\mathsf{LWR}_{n,p,q,\chi}$ distribution. Otherwise, if the $\mathcal{A}$ outputs 0 (mean that the input pair is from a uniform distribution over $R_p \times R_p$), then Sim also outputs 0 to mean that the pair (a, b) is from uniform distribution over $R_q \times R_p$.

Note that showing the following two statements suffices for the lemma: (1) if the input pair (a, b) is from uniform distribution over $R_q \times R_p$, then from $\mathcal{A}$'s view the pair $(\lceil a \rfloor_{q\to p}, b)$ is from the uniform distribution over $R_p \times R_p$, (2) if the input pair (a, b) is from $R\text{-}\mathsf{LWR}_{n,p,q,\chi}$ distribution, then from $\mathcal{A}$'s view the pair $(\lceil a \rfloor_{q\to p}, b)$ is a sample from the distribution $R\text{-}\mathsf{LWR}\text{-}\mathsf{AE}_{n,p,q,\chi}$.

The statement (1) is straight from the fact the $2p|q$ and the definition of the rounding function $\lceil * \rfloor_{q\to p}$, namely the rounding procedure maps a uniform element in R_q to a uniform element in R_p.

Next we show (2), if $b = \lceil a \cdot s \rfloor_{q\to p}$ for the $a \xleftarrow{\$} R_q$ and an element $s \in R_q$, then we prove that $b = \lceil (\frac{q}{p}a' + u) \cdot s \rfloor_{q\to p}$ for a uniform $a' := \lceil a \rfloor_{q\to p}$, and a uniform ring element $u = a - \frac{q}{p}a'$ whose coefficients are in $[\frac{q}{2p}, \frac{q}{2p})$, where the uniformity is taken over the uniformity of $a \in R_q$. Note that the uniformity of a' is by a same argument as the argument of the statement (1), and it's easy to see that the coefficients of u in $[\frac{q}{2p}, \frac{q}{2p})$. Therefore, we now show uniformity of u. In other words, for any $u_0 \in [\frac{q}{2p}, \frac{q}{2p})$, we show that $\mathsf{Pr}_a[u = u_0] = \left(\frac{p}{q}\right)^n$, that is

$$\begin{aligned}\mathsf{Pr}_a[u = u_0] &= \mathsf{Pr}_a[a = \frac{q}{p}a' + u_0] \\ &= \sum_{a_p \in R_p} \mathsf{Pr}_a[a = \frac{q}{p}a' + u_0 \wedge a' = a_p] \\ &= \sum_{a_p \in R_p} \frac{1}{q^n} = \left(\frac{p}{q}\right)^n,\end{aligned}$$

where the second equality is by the union bound; the third equality is by the uniformity of a over R_q. This completes the proof. □

A.2 Proof of Lemma 3.7

Proof. We show the lemma by two steps: (1) we show that the advantage of $\mathcal{A}$ in Game_2 is no larger than that of an intermediate game Game'_3 we introduced, and (2) from the view of $\mathcal{A}$ the two games Game'_3 and Game_3 is indistinguishable if the $R\text{-}\mathsf{LWR}_{2n,p,q,\mathsf{Bin}_\beta}$ problem is hard. The difference between the Game_2 and the intermediate game Game'_3 is the way they generate b and c_1 as follow

$$b \xleftarrow{\$} R_p, c_1 = \lceil (\frac{q}{p}b + u) \cdot r \rfloor_{q \to p}, \text{ and } b \xleftarrow{\$} R_q, c_1 = \lceil b \cdot r \rfloor_{q \to p}$$

Note that (1) is follows from the Lemma 3.4 that the advantage of $\mathcal{A}$ in Game_2 is no larger than that of Game'_3. Furthermore, an analogue reduction as in the proof of Lemma 3.6 shows (2). This completes the proof.

□

A.3 Proof of Lemma 3.6

Proof. The Game_2 is different from the Game_1 by the way the pk generated, that b is generated by computing the rounding function $\lceil a \cdot s \rfloor_{q \to p}$ for some secret $s \in R_q$, but it is sampled uniformly over the ring R_p in the Game_2. The lemma follows if from the view of $\mathcal{A}$ the two games are indifferent. We show this by contradiction. Assume that $\mathcal{A}$ can distinguish these two games, then we construct an algorithm Sim as follows that can solve the $R\text{-}\mathsf{LWR}_{n,p,q,\mathsf{Bin}_\beta}$.

$\mathsf{Sim}(a, b)$

On input the pair $(a, b) \in R_q \times R_p$ of $R\text{-}\mathsf{LWR}_{n,p,q,\mathsf{Bin}_\beta}$ challenge, it simulate every step in Game_1(or Game_2) for $\mathcal{A}$ except the step2 and step 4 that it uses (a, b) instead of sampling or computing them.

Note that if the pair (a, b) is from $R\text{-}\mathsf{LWR}_{n,p,q,\mathsf{Bin}_\beta}$, then Sim exactly simulated the Game_1 for $\mathcal{A}$, if the pair (a, b) is from uniform distribution over $R_q \times R_p$, then Sim exactly simulated the Game_2 for $\mathcal{A}$. Therefore, if $\mathcal{A}$ can distinguishes the two games with noticeable probability, then Sim can solve $R\text{-}\mathsf{LWR}_{n,p,q,\mathsf{Bin}_\beta}$ problem with same probability. This is contradicts the lemma assumption that $R\text{-}\mathsf{LWR}_{n,p,q,\mathsf{Bin}_\beta}$ problem is hard. Thus from the view of $\mathcal{A}$ the two games are indifferent, and the lemma follows. □

References

1. NIST (2020). https://csrc.nist.gov/projects/post-quantum-cryptography
2. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: 28th ACM STOC, pp. 99–108. ACM Press (May 1996)
3. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange - a new hope. In: Holz, T., Savage, S. (eds.) USENIX Security 2016, pp. 327–343. USENIX Association (August 2016)
4. Alperin-Sheriff, J., Apon, D.: Dimension-preserving reductions from LWE to LWR. IACR Cryptol. ePrint Arch. **2016**, 589 (2016)
5. Alwen, J., Krenn, S., Pietrzak, K., Wichs, D.: Learning with rounding, revisited. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 57–74. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_4
6. Avanzi, R., et al.: CRYSTALS-kyber. submission to the NIST post-quantum cryptography standardization project. NIST National Institute of Standards and Technology (2020)
7. Baan, H., et al.: Round5: compact and fast post-quantum public-key encryption. In: Ding, J., Steinwandt, R. (eds.) PQCrypto 2019. LNCS, vol. 11505, pp. 83–102. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-25510-7_5
8. Bai, S., Langlois, A., Lepoint, T., Stehlé, D., Steinfeld, R.: Improved security proofs in lattice-based cryptography: using the Rényi divergence rather than the statistical distance. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 3–24. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_1
9. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 719–737. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_42
10. Bogdanov, A., Guo, S., Masny, D., Richelson, S., Rosen, A.: On the hardness of learning with rounding over small modulus. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9562, pp. 209–224. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49096-9_9
11. Bos, J.W., et al.: Frodo: take off the ring! practical, quantum-secure key exchange from LWE. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016, pp. 1006–1018. ACM (2016)
12. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC, pp. 575–584. ACM Press (June 2013)
13. Chung, C.M., Hwang, V., Kannwischer, M.J., Seiler, G., Shih, C., Yang, B.: NTT multiplication for NTT-unfriendly rings new speed records for saber and NTRU on cortex-m4 and AVX2. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2021**(2), 159–188 (2021)
14. Dachman-Soled, D., Ducas, L., Gong, H., Rossi, M.: LWE with side information: attacks and concrete security estimation. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020. LNCS, vol. 12171, pp. 329–358. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-56880-1_12
15. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_34

16. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017. LNCS, vol. 10677, pp. 341–371. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70500-2_12
17. Jan-Pieter D'Anvers, S.S.R., Karmakar, A., Vercauteren, F.: SABER: Submission to the NIST post-quantum cryptography standardization project. NIST National Institute of Standards and Technology (2020)
18. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: IND-CCA-secure key encapsulation mechanism in the Quantum Random Oracle Model, revisited. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10993, pp. 96–125. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96878-0_4
19. Jin, Z., Zhao, Y.: Optimal key consensus in presence of noise. Cryptology ePrint Archive, Report 2017/1058 (2017). http://eprint.iacr.org/2017/1058
20. Liu, F.-H., Wang, Z.: Rounding in the rings. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020. LNCS, vol. 12171, pp. 296–326. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-56880-1_11
21. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_1
22. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Mitzenmacher, M. (ed.) 41st ACM STOC, pp. 333–342. ACM Press (May/June 2009)
23. Peikert, C.: Lattice cryptography for the internet. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 197–219. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11659-4_12
24. Peikert, C., Regev, O., Stephens-Davidowitz, N.: Pseudorandomness of ring-LWE for any ring and modulus. In: Hatami, H., McKenzie, P., King, V. (eds.) 49th ACM STOC, pp. 461–473. ACM Press (June 2017)
25. Poppelmann, T., et al.: NewHope - submission to the NIST post-quantum cryptography standardization project. NIST National Institute of Standards and Technology (2019)
26. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC, pp. 84–93. ACM Press (May 2005)
27. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: 35th FOCS, pp. 124–134. IEEE Computer Society Press (November 1994)
28. Zhang, J., Zhang, Z., Ding, J., Snook, M., Dagdelen, Ö.: Authenticated key exchange from ideal lattices. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 719–751. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_24
29. Zhang, Z., et al.: NTRU - technical report, national institute of standards and technology. NIST National Institute of Standards and Technology (2020)