Compact and Flexible KEM from Ideal Lattice

Zhengzhong Jin, Shiyu Shen, Yunlei Zhao

***Abstract*—A remarkable breakthrough in mathematics in recent years is the proof of the long-standing conjecture: sphere packing in the E_8 lattice is optimal in the sense of the best density for sphere packing in $\mathbb{R}^8$. In this work, we design a mechanism for asymmetric key consensus from noise (AKCN), referred to as AKCN-E8, for error correction and key consensus. As a direct application, we present a practical key encapsulation mechanism (KEM) from the ideal lattice based on the ring learning with errors (RLWE) problem.**

Compared with NewHope-KEM that was the second round candidate of the National Institute of Standards and Technology (NIST) post-quantum cryptography (PQC) standardization, our AKCN-E8 KEM scheme overcomes some limitations and shortcomings of NewHope-KEM. Compared with some other dominating KEM schemes based on the variants of LWE, specifically Kyber and Saber, AKCN-E8 has a comparable performance but enjoys much flexible shared-key sizes. Specifically, the key encapsulated by AKCN-E8-512 (resp., 768, 1024) has the size of 256 (resp., 384, 512) bits. Flexible key size renders us stronger security against quantum attacks, more powerful and economic ability of key transportation, and better matches the demand in interactive protocols like TLS where parties need to negotiate the security parameters including the shared key length.

***Index Terms*—post-quantum cryptography, error correction, lattice, key encapsulation mechanism, ring learning with errors problem.**

I. Introduction

Advancements in quantum computing have spurred the development of new public-key cryptographic primitives that are conjectured to be secure against quantum attacks. One promising class of these primitives is based on lattices, leading to key encapsulation mechanisms (KEM) based on the *learning with errors* (LWE) problem [31]. For cryptographic usage, compared with the classic hard lattice problems such as SVP and CVP, the *learning with error* (LWE) problem, and its variant *learning with rounding* (LWR), are proven to be much more versatile [36]. Nevertheless, LWE-based cryptosystems are usually less efficient, which was then resolved by the introduction of the ring-LWE (RLWE) problem [27] from ideal lattice. Among RLWE-based asymmetric primitives, NewHope-KEM [34] is one of the prominent KEM schemes, which is a variant of NewHope-Usenix [1] (winner of the 2016 Internet Defense Prize), and was a candidate in the second round of the NIST post-quantum cryptography (PQC) standardization competition.

In this work, we review the modular and generalized framework, explicitly proposed in [22, 23], for designing and analyzing KEM schemes from LWE and its variant (in particular, RLWE). This modular and generalized framework brings us to focus on one key building block for achieving KEMs from LWE and its variants, which is referred to as *asymmetric key consensus* (AKC). Putting into this framework, the underlying (one-dimensional) AKC mechanisms proposed in [25, 27, 36] encode one key bit per polynomial coefficient.

One-dimensional AKC was further optimized in [22, 23]. The work [35] extended one-dimensional reconciliation mechanisms into multi-dimensional ones based on the lattice code in D_2 and D_4. where one key bit is encoded into two (resp., four) polynomial coefficients by using the D_2 (resp., D_4) code. This multi-dimensional approach can allow either to improve the security of the resulting scheme or to decrease the probability of decryption failures. The D_4 code was adapted into key exchange scheme in [1] and later into KEM schemes in [22, 34].

Newhope-KEM has two variants based on the polynomial dimension $n = 512$ or 1024, referred to as NewHope-512 and NewHope-1024. Specifically, in order to derive the 256-bit shared-key, NewHope-512 (resp., NewHope-1024) uses the D_2 (resp., D_4) lattice code. Note that, on the same polynomial dimension, the D_2 lattice code can derive more bits of shared-key but at the cost of relatively higher error probability, while the D_4 code has a better error correction ability but with the size of shared-key halved. This causes NewHope-KEM to be less modular or flexible. For protocol simplicity, modularity and flexibility, we may hope for a new code mechanism that combines, in essence, the advantages of both the D_2 code and the D_4 code, while saving from or alleviating the disadvantages of them both.

A. Our Contributions

The encoding and decoding algorithms of E_8 were proposed by Conway and Sloane [7]. Recently, a remarkable breakthrough in mathematics is the proof of the long-standing conjecture: sphere packing in the E_8 lattice is optimal in the sense of the best density for packing in $\mathbb{R}^8$ [41]. However, to apply the algorithms of [7] to the KEM setting, we need to specify a one-to-one mapping from binary strings to lattice points in E_8.

This work was supported in part by the National Natural Science Foundation of China under Grant U1536205, Grant 61472084, Grant 61972094, Grant 62032005, Grant 61972159, and Grant 61572198; in part by the Fundamental Research Funds for the Central Universities; in part by the National Key Research and Development Program of China under Grant 2017YFB0802000; in part by the Shanghai Innovation Action Project under Grant 16DZ1100200; in part by the Shanghai Science and Technology Development Funds under Grant 16JC1400801; in part by the Technical Standard Project of Shanghai Scientific and Technological Committee under Grant 21DZ2200500; and in part by the Shandong Provincial Key Research and Development Program of China under Grant 2017CXG0701 and Grant 2018CXGC0701. (Corresponding author: Yunlei Zhao.)

Z.Z. Jin is with the Department of Computer Science, Fudan University, China, and the Department of Computer Science, Johns Hopkins University, USA.

S.Y. Shen and Y.L. Zhao are with the Department of Computer Science, Fudan University, China.

A natural way to specify such a mapping is to choose a base for the lattice E_8. Then, transforming the lattice points to the binary strings may involve Gaussian elimination. Compared to this method, our encoding and decoding algorithms integrate the coding of E_8 and the mapping from binary strings to E_8 together. This improves the efficiency by avoiding Gaussian elimination. Finally, we adapt the integrated E_8 code into the KEM setting, by combining it with the AKCN scheme of [22]. The resultant code is referred to as AKCN-E8.

As a direct application of the AKCN-E8 code, we present a practical KEM scheme based on the RLWE assumption, which is referred to as AKCN-E8-KEM. Compared with NewHope-KEM [34], our AKCN-E8-KEM has the following advantages:

- Modular and unified constructions: The same E_8 code is employed for all the three dimensions $n = 512$, $n = 768$ and $n = 1024$.
- For the recommended case of $n = 1024$, the size of shared-key is doubled: 512 bits of AKCN-E8 versus 256 bits of NewHope.
- At the same security level and the same length of shared-key, AKCN-E8 can enjoy both more compact ciphertext size and lower error probability.
- More flexible parameter selection for tradeoffs among security, ciphertext size and error probability.

NewHope-KEM does not provide a set of parameters for the dimension $n = 768$. In comparison, the KEM schemes based on the variants of LWE currently in the third round of NIST PQC standardization, specifically Kyber that is based on module-LWE (MLWE) and Saber that is based on module-LWR (MLWR), do provide this level of parameters. One reason is that the standard NTT technique requires that n be power-of-two and $q \mod 2n = 1$. Recent advances on the variants of NTT [2, 24, 29, 42, 43] allow us to choose NTT-friendly parameters in a more flexible way. In this work, we provide the parameter set of $n = 768$ and $q = 7681$ for AKCN-E8 (referred to as AKCN-E8-768), by employing the P3-NTT proposed in [24] that enjoys better paralleability and modularity. We then give detailed comparisons among AKCN-E8, Kyber and Saber. Briefly speaking, in comparison with Kyber and Saber, AKCN-E8 has a comparable performance but enjoys much flexible sizes of the shared-key to be encapsulated. Specifically, the key encapsulated by AKCN-E8-512 (resp., 768, 1024) has the size of 256 (resp., 384, 512) bits, compared to the fixed key size of 256 bits for Kyber and Saber.

We make a comprehensive analysis of the error probability of the AKCN-E8-KEM scheme. As a by-product result, we also show that the errors in different positions for RLWE-based KEM schemes are independent when the polynomial dimension tends to be infinity. This is a complementary result to the dependency result for concrete parameters presented at PQC 2019 [11], which might also be of independent interest.

For performance benchmarks and comparisons with NewHope, Kyber and Saber, we provide thorough implementations of the proposed AKCN-E8 scheme, covering the parameters $n \in \{512, 768, 1024\}$ and $q \in \{12289, 3329, 7681\}$. The implementation codes are available from http://github.com/AKCN-E8.

On the importance and desirability of flexible shared-key size. The shared-key size in bits for both NewHope-512 and NewHope-1024 is 256. Also, the shared-key size for both Kyber and Saber is fixed to be 256 bits, which is intrinsic to the dimension of the underlying module lattice and is inflexible to change (for example, it is uneasy to employ a module lattice of dimension $n = 384$). In comparison, the shared-key size of AKCN-E8-512 (resp., -768, -1024) is 256 (resp., 384, 512) bits. Here, we would like to highlight the importance and desirability of larger shared-key size.

- Doubling the shared-key size means more powerful and economic ability of key transportation, at about the same level of security and bandwidth.
- A typical application of KEM in practice is to encapsulate a pair of keys (K_1, K_2), where K_1 (resp., K_2) is used as the key for symmetric-key encryption like AES (resp., for message authentication code like HMAC). When instantiated with AES-256 (resp., AES-192), each of K_1 and K_2 has size of 256 (resp., 192) bits. In these application scenarios, running a KEM scheme (encapsulating key of 192 or 256 bits) twice is much less efficient than running a KEM scheme (encapsulating key of 384 or 512 bits) once.
- For some application scenarios demanding critical security guarantees, symmetric-key cryptographic primitives of larger key size (particularly, key size of 512 bits) are already in use in practice.
- Fixing key size for different security levels is less flexible. A more flexible and desirable way is to allow users to negotiate the key sizes according to different security levels and application scenarios. For example, according to different security levels (specifically, 128, 192, 256 bit classic security), in TLS 1.3 [38] it mandates three options for the master secrecy size: 256, 384 and 512, by negotiating and employing the secp256r1, secp384r1 and secp512r1 curves respectively.
- Doubling the shared-key size is important for the targeted security level against Grover's search algorithm, and against the possibility of more sophisticated quantum cryptanalysis in the long run. Note that for all the protocols of NewHope-1024, AKCN-E8-1024, Kyber-1024 and FireSaber, their target security level is about 230-bit post-quantum security. Even if the underlying lattice hard problems provide this level of hardness, the 256-bit shared-key may not. Though the standardization of post-quantum symmetric key cryptography is not considered yet, it is expected that the key size will increase to remain the same security level in the post-quantum era. For example, the updated quantum analysis on AES [20] overall reduces the original estimate of quantum cost in bits against AES (specified in the call for proposals of NIST PQC standardization [31]) between 11 and 13, and this line of research is quite active now.

B. Related Work

The Leech lattice is also proven to be the densest for sphere packing in dimension 24 [6], and has already been used for error correction in communication protocols [8, 40], for example in the IEEE 802.11a WLAN standard https://standards.ieee.org/standard/802_11-2016.html. On the one hand, its encoding and decoding are more complex and less efficient than the AKCN-E8 code. On the other hand, and more importantly, it is difficult to find parameters of RLWE [33], since it is a 24-dimension lattice. For RLWE-based cryptosystems, we usually use number-theoretic transform (NTT) algorithms to speed up the polynomial multiplications. The NTT algorithms can make the most use of the computational resource when the dimension of RLWE is a power of 2. However, one cannot hope for setting the parameter n to be a power of 2 and a multiple of 24 at the same time. The same issue also occurs when setting the key length for Leech lattice, since the key size usually will be a multiple of 12. In comparison, the E_8 lattice doesn't have the aforementioned problems.

The recommended parameter set of NewHope-KEM aims for about 256-bit classic security and about 230-bit post-quantum security (pq-sec). For the KEM proposals in the second round of NIST PQC standardization, the LAC algorithm [26] is another RLWE-based KEM scheme. To our knowledge, NewHope-KEM and LAC are the only two RLWE-based KEM proposals left in the second round of NIST PQC standardization. LAC uses a different approach for building KEM from RLWE: it uses a small $q = 251$ that is not NTT-friendly, and uses an error correction code (ECC) to lower the failure probability. LAC also proposes parameters for about 256-bit classic security but with a relatively higher failure probability. Compared with the lattice code based approach of NewHope and this work, the KEM schemes based on the ECC-based approach is usually more complex, and is harder for constant-time implementations. Recently, it is also shown that the LAC algorithm does not achieve its claimed security level [17].

In the second round of NIST PQC standardization, there are only NewHope and LAC that are based on the RLWE problem. Some other well-known lattice-based KEM candidates, e.g., Kyber [5] and Saber [10], are based on the module lattice. The underlying polynomials of Kyber and Saber are of dimension $n = 256$. This leaves no space for employing lattice codes or ECC codes for extra error correction, besides direct error correction with the AKC mechanism. In addition, Saber uses a special modulus q that is a power-of-two, and as a consequence the NTT technique cannot be used. Due to the protocol simplicity, we suggest Kyber can be more efficient than NewHope and AKCN-E8. But both Saber and Kyber are hard to derive 512-bit shared-key, even if the underlying hard problem provides about 256-bit post-quantum security. So, in general, AKCN-E8 is incomparable with Kyber and Saber.

II. Preliminaries

A string or value α means a binary one, and $|\alpha|$ is its binary length. For any real number x, $\lfloor x \rfloor$ denotes the largest integer that is less than or equal to x, and $\lfloor x \rceil = \lfloor x + 1/2 \rfloor$. For any positive integers a and b, denote by $\mathsf{lcm}(a, b)$ the least common multiple of them. For any $i, j \in \mathbb{Z}$ such that $i < j$, denote by $[i, j]$ the set of integers $\{i, i+1, \cdots, j-1, j\}$. For any positive integer t, we let $\mathbb{Z}_t$ denote $\mathbb{Z}/t\mathbb{Z}$. The elements of $\mathbb{Z}_t$ are represented, by default, as $[0, t-1]$. Nevertheless, sometimes, $\mathbb{Z}_t$ is explicitly specified to be represented as $[-\lfloor (t-1)/2 \rfloor, \lfloor t/2 \rfloor]$.

If $\mathcal{S}$ is a finite set, then $|\mathcal{S}|$ is its cardinality, and $x \leftarrow \mathcal{S}$ is the operation of picking an element uniformly at random from $\mathcal{S}$. For two sets $A, B \subseteq \mathbb{Z}_q$, define $A + B \triangleq \{a + b | a \in A, b \in B\}$. For an addictive group $(G, +)$, an element $x \in G$ and a subset $S \subseteq G$, denote by $x + S$ the set containing $x + s$ for all $s \in S$. For a set S, denote by $\mathcal{U}(S)$ the uniform distribution over S. For any discrete random variable X over $\mathbb{R}$, denote $\mathsf{Supp}(X) = \{x \in \mathbb{R} \mid \Pr[X = x] > 0\}$.

We use standard notations and conventions below for writing probabilistic algorithms, experiments and interactive protocols. If $\mathcal{D}$ denotes a probability distribution, $x \leftarrow \mathcal{D}$ is the operation of picking an element according to $\mathcal{D}$. If α is neither an algorithm nor a set, $x \leftarrow \alpha$ is simple assignment statement. If A is a probabilistic polynomial-time (PPT) algorithm, then $A(x_1, x_2, \cdots; r)$ is the result of running A on inputs $x_1, x_2, \cdots$ and coins r. We let $y \leftarrow A(x_1, x_2, \cdots)$ denote the experiment of picking r at random and letting y be $A(x_1, x_2, \cdots; r)$. By $\Pr[R_1; \cdots; R_n : E]$ we denote the probability of event E, after the ordered execution of random processes $R_1, \cdots, R_n$. A function $f(\lambda)$ is *negligible*, if for every $c > 0$ there exists an λ_c such that $f(\lambda) < 1/\lambda^c$ for all $\lambda > \lambda_c$.

A. Key Encapsulation Mechanism (KEM)

We review the definition of KEM given in [12, 18]. A key encapsulation mechanism $\mathsf{KEM} = (\mathsf{KeyGen}, \mathsf{Encaps}, \mathsf{Decaps})$ consists of three algorithms. On a security parameter κ, the PPT key generation algorithm KeyGen outputs a key pair (pk, sk), where pk also defines a finite key space $\mathcal{K}$. The PPT encapsulation algorithm Encaps, on input pk, outputs a tuple (K, c) where c is said to be an encapsulation of the key K which is contained in key space $\mathcal{K}$. The deterministic polynomial-time decapsulation algorithm Decaps, on input sk and an encapsulation c, outputs either a key $K := \mathsf{Decaps}(sk, c) \in \mathcal{K}$ or a special symbol $\perp \notin \mathcal{K}$ to indicate that c is not a valid encapsulation. We call KEM *δ-correct* if

$$\begin{aligned}\Pr[\mathsf{Decaps}(sk, c) \neq K | (pk, sk) \leftarrow \mathsf{KeyGen}(1^\kappa);\\ (K, c) \leftarrow \mathsf{Encaps}(pk)] \leq \delta.\end{aligned}$$

The security notion, indistinguishability under chosen ciphertext attacks (CCA), is defined w.r.t. Fig. 1. For any PPT adversary $\mathcal{A}$, define its CCA-advantage as $Adv_{KEM}^{CCA}(\mathcal{A}) := |Pr[\textbf{GAME}\ \mathsf{CCA}\ \text{outputs}\ 1]| - 1/2|$. We say the KEM scheme is CCA-secure, if for any sufficiently larger security parameter and any PPT adversary $\mathcal{A}$, $Adv_{KEM}^{CCA}(\mathcal{A})$ is negligible.

B. Public-Key Encryption (PKE)

We review the definition of PKE given in [16, 18]. A public-key encryption scheme is given by a triple of algorithms, $\mathsf{PKE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, where for every sufficiently large $\kappa \in \mathbb{N}$.

- KeyGen, the key-generation algorithm, is a probabilistic polynomial-time (in κ) algorithm which on input 1^κ outputs a pair of strings, (pk, sk), called the public and secret keys, respectively. This experiment is written as $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\kappa)$.

```
GAME  IND-CCA                                  DECAPS($c \neq c^*$)
$(pk, sk) \leftarrow$ Gen                      $K :=$ Decaps$(sk, c)$
$b \xleftarrow{\$} \{0,1\}$                    return  $K$
$(K_0^*, c^*) \leftarrow$ Encaps$(pk)$
$K_1^* \xleftarrow{\$} \mathcal{K}$
$b' \leftarrow \mathsf{A}^{\text{DECAPS}}(c^*, K_b^*)$
return  $[b' = b]$
```

Fig. 1: CCA game for KEM

- $\mathcal{E}$, the encryption algorithm, is a probabilistic polynomial-time (in κ) algorithm that takes public key pk and message M from the message space MSP, draws coins r uniformly from coin space COIN, and produces ciphertext $C := \mathcal{E}_{pk}(M; r)$. This experiment is written as $C \leftarrow \mathcal{E}_{pk}(x)$.
- $\mathcal{D}$, the decryption algorithm, is a deterministic polynomial-time (in κ) algorithm that takes secret key sk and ciphertext $C \in \{0,1\}^*$, and returns message $M \in$ MSP.

We say a PKE scheme is δ-correct, if for every sufficiently large $\kappa \in \mathbb{N}$, every (pk, sk) generated by KeyGen(1^κ) and every $M \in$ MSP, we always have $\mathbf{E}[\max_{M \in \text{MSP}} \Pr[\mathcal{D}_{sk}(\mathcal{E}_{pk}(M)) \neq M]] \leq \delta$.

Definition 1 (CCA-security)**.** *Let PKE $= (\textit{KeyGen}, \mathcal{E}, \mathcal{D})$ be an asymmetric encryption scheme, and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary for PKE. For $\kappa \in \mathbb{N}$, define the following CCA-advantage:*

$$\begin{aligned}\mathbf{Adv}_{\mathcal{A}}^{\mathsf{CCA}}(\kappa) = 2 \cdot \Pr[&(pk, sk) \leftarrow \textit{KeyGen}(1^\kappa);\\ &(M_0, M_1, st) \leftarrow \mathcal{A}_1^{\mathcal{D}_{sk}}(pk);\\ &b \leftarrow \{0,1\}; C^* \leftarrow \mathcal{E}_{pk}(M_b):\\ &\mathcal{A}_2^{\mathcal{D}_{sk}}(C^*, st) = b] - 1.\end{aligned}$$

We say that the PKE scheme is CCA-secure, if for every sufficiently large security parameter κ, and PPT adversary $\mathcal{A}$, its CCA-advantage $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{CCA}}$ is negligible in κ. We say the PKE scheme is secure against chosen plaintext attacks (CPA-secure, for short), if the advantage of $\mathcal{A}$ is negligible when the access to the decryption oracle $\mathcal{D}_{sk}$ is denied.

C. The LWE, and Ring-LWE (RLWE) problems

Given positive *continuous* $\sigma > 0$, define the real Gaussian function $\rho_\sigma(x) \triangleq \exp(-x^2/2\sigma^2)/\sqrt{2\pi\sigma^2}$ for $x \in \mathbb{R}$. Let $D_{\mathbb{Z},\sigma}$ denote the one-dimensional *discrete* Gaussian distribution over $\mathbb{Z}$, which is determined by its probability density function $D_{\mathbb{Z},\sigma}(x) \triangleq \rho_\sigma(x)/\rho_\sigma(\mathbb{Z}), x \in \mathbb{Z}$. Finally, let $D_{\mathbb{Z}^n,\sigma}$ denote the n-dimensional *spherical* discrete Gaussian distribution over $\mathbb{Z}^n$, where each coordinate is drawn *independently* from $D_{\mathbb{Z},\sigma}$.

Given positive integers n and q that are both polynomials in the security parameter λ, an integer vector $\mathbf{s} \in \mathbb{Z}_q^n$, and a probability distribution χ on $\mathbb{Z}_q$, let $A_{q,\mathbf{s},\chi}$ be the distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, and an error term $e \leftarrow \chi$, and outputting the pair $(\mathbf{a}, b = \mathbf{a}^T\mathbf{s} + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The error distribution χ is typically taken to be the discrete Gaussian probability distribution $D_{\mathbb{Z},\sigma}$ defined previously; However, as suggested in [4] and as we shall see in Section V, other alternative distributions of χ can be taken. Briefly speaking, the (decisional) *learning with errors* (LWE) assumption [36] says that, for sufficiently large security parameter λ, no probabilistic polynomial-time (PPT) algorithm can distinguish, with non-negligible probability, $A_{q,\mathbf{s},\chi}$ from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$. This holds even if $\mathcal{A}$ sees polynomially many samples, and even if the secret vector $\mathbf{s}$ is drawn randomly from χ^n [3].

For the positive integer m that is polynomial in the security parameter λ, let $n \triangleq \varphi(m)$ denote the toties of m, and $\mathcal{K} \triangleq \mathbb{Q}(\zeta_m)$ be the number field obtained by adjoining an abstract element ζ_m satisfying $\Phi_m(\zeta_m) = 0$, where $\Phi_m(x) \in \mathbb{Z}[x]$ is the m-th cyclotomic polynomial of degree n. Moreover, let $\mathcal{R} \triangleq \mathcal{O}_{\mathcal{K}}$ be the ring of integers in $\mathcal{K}$. Finally, given a positive prime $q = \text{poly}(\lambda)$ such that $q \equiv 1 \pmod m$, define the quotient ring $\mathcal{R}_q \triangleq \mathcal{R}/q\mathcal{R}$.

We briefly review the RLWE problem, and its hardness result [13, 27, 28]. In this work, we focus on a *special* case of the RLWE problem defined in [27]. Let $n \geq 16$ be a power-of-two and $q = \text{poly}(\lambda)$ be a positive prime such that $q \equiv 1 \pmod{2n}$. Given $\mathbf{s} \leftarrow \mathcal{R}_q$, a sample drawn from the RLWE distribution $A_{n,q,\sigma,\mathbf{s}}$ over $\mathcal{R}_q \times \mathcal{R}_q$ is generated by first choosing $\mathbf{a} \leftarrow \mathcal{R}_q, \mathbf{e} \leftarrow D_{\mathbb{Z}^n,\sigma}$, and then outputting $(\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + \mathbf{e}) \in \mathcal{R}_q \times \mathcal{R}_q$. Roughly speaking, the (decisional) RLWE assumption says that, for sufficiently large security parameter λ, no PPT algorithm $\mathcal{A}$ can distinguish, with non-negligible probability, $A_{n,q,\sigma,\mathbf{s}}$ from the uniform distribution over $\mathcal{R}_q \times \mathcal{R}_q$. This holds even if $\mathcal{A}$ sees polynomially many samples, and even if the secret $\mathbf{s}$ is drawn randomly from the same distribution of the error polynomial $\mathbf{e}$ [3, 13]. Moreover, as suggested in [1], alternative distributions for the error polynomials can be taken for the sake of efficiency while without essentially reducing security.

Recently, a polynomial-time (quantum) reduction from worst-case ideal lattice problems *directly* to the decision version of Ring-LWE is presented in [32]. In particular, the reduction works for any modulus and any number field. Besides the above special version of the RLWE problem [27], another suggested version of the RLWE problem is defined over the polynomial ring $\mathcal{R}_n = \mathbb{Z}[x]/\Phi_{n+1}(x)$, where $n+1$ is a safe prime and $\Phi_{n+1}(x) = x^n + x^{n-1} + \cdots + x + 1$ is the $(n+1)$-th cyclotomic polynomial. This ring has a wider range of n to choose from.

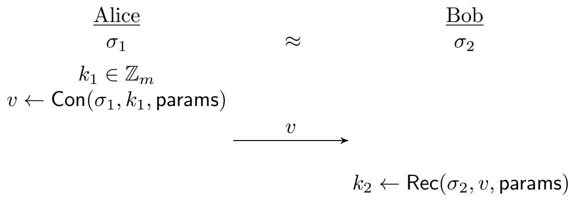
Alice
σ_1
$\approx$
Bob
σ_2
$k_1 \in \mathbb{Z}_m$
$v \leftarrow \mathsf{Con}(\sigma_1, k_1, \mathsf{params})$
v
$k_2 \leftarrow \mathsf{Rec}(\sigma_2, v, \mathsf{params})$

Fig. 2: Depiction of AKC

III. A Modular and Generalized Framework for PKE/KEM from Ring-LWE

A. Building Block: Asymmetric Key Consensus

Before presenting the definition of asymmetric key consensus (AKC) scheme, we first introduce a new function $|\cdot|_q$ relative to a positive integer $q \geq 1$: $|x|_q = \min\{x \bmod q, q - x \bmod q\}, \quad \forall x \in \mathbb{Z}$, where the result of modular operation is represented in $\{0, ..., (q-1)\}$. For instance, $|-1|_q = \min\{-1 \mod q, (q+1) \mod q\} = \min\{q-1, 1\} = 1$. For any $\mathbf{x} = (x_0, x_1, x_2, x_{\mu-1})^T \in \mathbb{Z}_q^\mu$, where μ is a positive integer, denote by $\|\mathbf{x}\|_{q,1}$ the sum $|x_0|_q + |x_1|_q + \cdots + |x_{\mu-1}|_q$.

Definition 2. *An asymmetric key consensus scheme $AKC = (\mathsf{params}, \mathsf{Con}, \mathsf{Rec})$ is specified as follows:*

- *$\mathsf{params} = (q, m, g, d, aux)$ denotes the system parameters, where q, $2 \leq m, g \leq q, 1 \leq d \leq \lfloor \frac{q}{2} \rfloor$ are positive integers, and aux denotes some auxiliary values that are usually determined by (q, m, g, d) and could be set to be empty.*
- *$\mathbf{v} \leftarrow \mathsf{Con}(\text{œ}_1, k_1, \mathsf{params})$: On input of $(\text{œ}_1 \in \mathbb{Z}_q^\mu, \mathbf{k}_1 \in \mathbb{Z}_m^{\mu'}, \mathsf{params})$, where μ and μ' are positive integers, the polynomial-time conciliation algorithm Con outputs the public hint $\mathbf{v} \in \mathbb{Z}_g^\mu$.*
- *$\mathbf{k}_2 \leftarrow \mathsf{Rec}(\text{œ}_2, \mathbf{v}, \mathsf{params})$: On input of $(\text{œ}_2 \in \mathbb{Z}_q^\mu, \mathbf{v} \in \mathbb{Z}_g^\mu, \mathsf{params})$, the* deterministic *polynomial-time algorithm Rec outputs $\mathbf{k}_2 \in \mathbb{Z}_m^{\mu'}$.*

Correctness: An AKC scheme is correct, *if it holds $\mathbf{k}_1 = \mathbf{k}_2$ for any $\text{œ}_1, \text{œ}_2 \in \mathbb{Z}_q^\mu$ such that $\|\text{œ}_1 - \text{œ}_2\|_{q,1} \leq d$.*
Security: An AKC scheme is secure, *if $\mathbf{v}$ is independent of $\mathbf{k}_1$ whenever œ_1 is uniformly distributed over $\mathbb{Z}_q^\mu$. Specifically, for arbitrary $\tilde{\mathbf{v}} \in \mathbb{Z}_g^\mu$ and arbitrary $\tilde{\mathbf{k}}_1, \tilde{\mathbf{k}}_1' \in \mathbb{Z}_m^{\mu'}$, it holds that $\Pr[\mathbf{v} = \tilde{\mathbf{v}} | \mathbf{k}_1 = \tilde{\mathbf{k}}_1] = \Pr[\mathbf{v} = \tilde{\mathbf{v}} | \mathbf{k}_1 = \tilde{\mathbf{k}}_1']$, where the probability is taken over $\text{œ}_1 \leftarrow \mathbb{Z}_q^\mu$ and the random coins possibly used by Con.*

B. CPA-Secure PKE from AKC

Denote by $(\lambda, n, q, \sigma, AKC)$ the system parameters, where λ is the security parameter. $q \geq 2$ is a positive prime number, σ parameterizes the discrete Gaussian distribution $D_{\mathbb{Z}^n, \sigma}$, n denotes the degree of polynomials in $\mathcal{R}_q$ where for simplicity we assume $\mu | n$, and Gen is a pseudorandom generator (PRG) generating $\mathbf{a} \in \mathcal{R}_q$ from a small seed $\mathsf{seed} \leftarrow \{0,1\}^\kappa$. Let $AKC = (\mathsf{params}, \mathsf{Con}, \mathsf{Rec})$ be a correct and secure AKC scheme, where $\mathsf{params} = (q, g, m, d)$. In this work, we mainly consider $m = 2$. The AKC-based PKE from RLWE is depicted in Figure 3 (page 6). Here, $(\mathsf{seed}, \mathbf{y}_1)$ serves as the public key, while $(\mathbf{y}_2, \mathbf{v})$ is the ciphertext. In the protocol description, for presentation simplicity, the Con and Rec functions are applied to polynomials, meaning they are applied to each group of μ coefficients respectively. For NewHope $\mu = 4$, while for AKCN-E8 $\mu = 8$. For presentation simplicity, we also referred to $\mathbf{k}_1 = \mathbf{k}_2$ as the shared-key.

It is well established that, under the assumptions that (1) the underlying AKC scheme is both correct and secure, and (2) the (decisional) RLWE is hard, the above modular construction of PKE scheme is CPA-secure [4, 22, 23, 25, 27, 36]. The above modular and generalized framework for CPA-secure PKE from LWE and its variants was explicitly proposed by Jin and Zhao [22], by explicitly defining and studying the underlying building tool AKC. All the previous works used AKC implicitly in a non-black-box way.

C. Transformation from CPA-PKE to CCA-KEM

There are well-established approaches from CPA-secure PKE to CCA-secure KEM [15, 16, 18, 19, 21, 39], with concrete security estimation in the quantum random oracle model (QROM). In this work, for presentation simplicity and ease of comparison, we use the same CCA transformation approach adopted by NewHope-KEM. The reader is referred to [34] for more details.

IV. Design and Analysis of AKCN-E8

According to the above modular and generalized framework from AKC to RLWE-based CPA and CCA secure KEMs, all left is to develop a practical AKC scheme, which is referred to AKCN-E8 to be developed and analyzed in this section. At the heart of AKCN-E8 is a novel lattice code in E_8.

We divide the coefficients of the polynomial $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ into $\hat{n} = n/8$ groups, where each group is composed of 8 coefficients. In specific, denote $R = \mathbb{Z}[x]/(x^8+1), R_q = R/qR, K = \mathbb{Q}[x]/(x^8+1)$ and $K_\mathbb{R} = K \otimes \mathbb{R} \simeq \mathbb{R}[x]/(x^8+1)$. Then the polynomial $\boldsymbol{\sigma}_1$ can be represented as $\boldsymbol{\sigma}_1(x) = \sigma_0(x^{\hat{n}}) + \sigma_1(x^{\hat{n}})x + \cdots + \sigma_{\hat{n}-1}(x^{\hat{n}})x^{\hat{n}-1}$, where $\sigma_i(x) \in R_q$ for $i = 0, 1, \ldots \hat{n}$. $\boldsymbol{\sigma}_2$ can be divided in the same way. Then we only need to construct the reconciliation mechanism for each $\sigma_i(x)$, and finally combine the keys together. To do this, we need to first introduce the lattice E_8 and its encoding and decoding.

We construct the lattice E_8 from the Extended Hamming Code in dimension 8, which is denoted as H_8 for presentation simplicity. H_8 refers to the 4-dimension linear subspace of 8-dimension linear space $\mathbb{Z}_2^8$.

$$H_8 = \{\mathbf{c} \in \mathbb{Z}_2^8 \mid \mathbf{c} = \mathbf{z}\mathbf{H} \bmod 2, \mathbf{z} \in \mathbb{Z}^4\}$$

where

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

The encoding algorithm is straightforward: given a 4-bit string $\mathbf{k}_1$, calculate $\mathbf{k}_1\mathbf{H}$. This operation can be done efficiently by bitwise operations. The complete algorithm is shown in Algorithm 1.[1]

Algorithm 1 AKCN-E8: Con with encoding in E_8

```
procedure: $Con(\sigma_1 \in \mathbb{Z}_q^8, \mathbf{k}_1 \in \mathbb{Z}_2^4, params)$
    $\mathbf{v} = \left\lfloor \frac{g}{q} \left( \boldsymbol{\sigma}_1 + \frac{q-1}{2} (\mathbf{k}_1 \mathbf{H} \bmod 2) \right) \right\rceil \bmod g^2$
    return $\mathbf{v}$
end procedure
```

[1] For simplicity, we assume q is a prime and directly use $\frac{q-1}{2}$ in Con (rather than $\lfloor q/2 \rceil$). The construction and analysis can be trivially changed to work with $\frac{q+1}{2}$ in Con. Also, when q is an even number (e.g., power-of-two), it should be $\frac{q}{2}$.

Initiator

$\mathsf{seed} \leftarrow \{0,1\}^{\kappa}$
$\mathbf{a} = \mathsf{Gen}(\mathsf{seed}) \in \mathcal{R}_q$
$\mathbf{x}_1, \mathbf{e}_1 \leftarrow D_{\mathbb{Z}^n,\sigma}$
$\mathbf{y}_1 = \mathbf{a} \cdot \mathbf{x}_1 + \mathbf{e}_1$

$$\xrightarrow{\mathsf{seed}, \mathbf{y}_1 \in \mathcal{R}_q}$$

Responder

$\mathbf{k}_2 \in \mathbb{Z}_m^{n/\mu}$
$\mathbf{a} = \mathsf{Gen}(\mathsf{seed})$
$\mathbf{x}_2, \mathbf{e}_2 \leftarrow D_{\mathbb{Z}^n,\sigma}$
$\mathbf{y}_2 = \lfloor \frac{2^t}{q}(\mathbf{a} \cdot \mathbf{x}_2 + \mathbf{e}_2) \rceil$
$\mathbf{e}_2' \leftarrow D_{\mathbb{Z}^n,\sigma}$
$\boldsymbol{\sigma}_2 = \mathbf{y}_1 \cdot \mathbf{x}_2 + \mathbf{e}_2' \in \mathcal{R}_q$
$\mathbf{v} \leftarrow \mathsf{Con}(\boldsymbol{\sigma}_2, \mathbf{k}_2, \mathsf{params})$

$$\xleftarrow{\mathbf{y}_2 \in \mathcal{R}_q, \mathbf{v} \in \mathcal{R}_g}$$

$\boldsymbol{\sigma}_1 = \lfloor \frac{q}{2^t} \mathbf{y}_2 \cdot \mathbf{x}_1 \rceil \in \mathcal{R}_q$
$\mathbf{k}_1 \leftarrow \mathsf{Rec}(\boldsymbol{\sigma}_1, \mathbf{v}, \mathsf{params})$

Fig. 3: Depiction of RLWE-based CPA-secure PKE from AKC

The decoding algorithm finds the solution of the closest vector problem (CVP) for the lattice E_8. For any given $\mathbf{x} \in \mathbb{R}^8$, CVP asks which lattice point in E_8 is closest to $\mathbf{x}$. Based on the structure of E_8, we propose an efficient decoding algorithm.

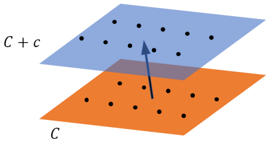

Fig. 4: Structure of E_8.

Let $C = \{(x_1, x_1, x_2, x_2, x_3, x_3, x_4, x_4) \in \mathbb{Z}_2^8 \mid x_1 + x_2 + x_3 + x_4 = 0 \bmod 2\}$. In fact, C is spanned by the up most three rows of $\mathbf{H}$. Hence, $E_8 = C \cup (C + \mathbf{c})$, where $\mathbf{c} = (0,1,0,1,0,1,0,1)$ is the last row of $\mathbf{H}$. For a given $\mathbf{x} \in \mathbb{R}^8$, to solve CVP of $\mathbf{x}$ in E_8, we solve CVP of $\mathbf{x}$ and $\mathbf{x} - \mathbf{c}$ in C, and then choose the one that has smaller distance. For a pictorial representation of E_8, refer to Figure 4.

Algorithm 2 AKCN-E8: Rec with decoding in E_8

```
procedure: $Rec(\sigma_2 \in \mathbb{Z}_q^8, \mathbf{v} \in \mathbb{Z}_g^8, params)$
    $\mathbf{k}_2 = \mathsf{Decode}_{\mathsf{E_8}}\left(\left\lfloor \frac{q}{g}\mathbf{v} \right\rceil - \boldsymbol{\sigma}_2\right)$
    return $\mathbf{k}_2$
end procedure
```

Then we consider how to solve CVP in C. For an $\mathbf{x} \in \mathbb{R}^8$, we choose $(x_1, x_2, x_3, x_4) \in \mathbb{Z}_2^4$, such that $(x_1, x_1, x_2, x_2, x_3, x_3, x_4, x_4)$ is closest to $\mathbf{x}$. However, $x_1 + x_2 + x_3 + x_4 \bmod 2$ may be equal to 1. In such cases, we choose the 4-bit string (x_1', x_2', x_3', x_4') such that $(x_1', x_1', x_2', x_2', x_3', x_3', x_4', x_4')$ is secondly closest to $\mathbf{x}$. Note that (x_1', x_2', x_3', x_4') has at most one-bit difference from (x_1, x_2, x_3, x_4). The detailed algorithm is depicted in Algorithm 3. Considering potential timing attack, all the "if" conditional statements can be implemented by constant time bitwise operations. In practice, Decode_C^{00} and Decode_C^{01} are implemented as two subroutines.

For Algorithm 3 (page 7), in Decode_{E_8}, we calculate $\mathsf{cost}_{i,b}$, where $i = 0, 1, \ldots, 7, b \in \{0,1\}$, which refer to the contribution to the total 2-norm when $x_i = b$. Decode_C^{00} solves the CVP in lattice C, and Decode_C^{01} solves the CVP in lattice $C + \mathbf{c}$. Then we choose the one that has smaller distance. $\mathsf{Decode}_C^{b_0 b_1}$ calculates the $k_i, i = 0,1,2,3$ such that $\frac{q-1}{2}(k_0 \oplus b_0, k_0 \oplus b_1, k_1 \oplus b_0, k_1 \oplus b_1, k_2 \oplus b_0, k_2 \oplus b_1, k_3 \oplus b_0, k_3 \oplus b_1)$ is closest to $\mathbf{x}$. We use min_d and min_i to find the second closest vector. Finally, we check the parity to decide which one should be returned.

The following theorem gives a condition of success of the encoding and decoding algorithm in Algorithm 1 and Algorithm 2. For simplicity, for any $\boldsymbol{\sigma} = (x_0, x_1, \ldots, x_7) \in \mathbb{Z}_q^8$, we define $\|\boldsymbol{\sigma}\|_{q,2}^2 = \sum_{i=0}^{7} |x_i|_q^2$.

Theorem 1. *If* $\|\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2\|_{q,2} \leq (q-1)/2 - \sqrt{2}\left(\frac{q}{g} + 1\right)$, *then* $\mathbf{k}_1$ *and* $\mathbf{k}_2$ *calculated by* Con *and* Rec *are equal.*

Proof. The minimal Hamming distance of the Extended Hamming code H_8 is 4. Hence, the minimal distance in the lattice we used is $\frac{1}{2}\sqrt{\left(\frac{q-1}{2}\right)^2 \times 4} = (q-1)/2$.

Algorithm 3 Decoding in E_8 and C

```
procedure $\mathsf{Decode}_{E_8}(\mathbf{x} \in \mathbb{Z}_q^8)$
    for $i = 0 \ldots 7$ do
        $\mathsf{cost}_{i,0} = |x_i|_q^2$
        $\mathsf{cost}_{i,1} = |x_i - \frac{q-1}{2}|_q^2$
    end for
    $(\mathbf{k}^{00}, \mathsf{TotalCost}^{00}) \leftarrow \mathsf{Decode}_C^{00}(\mathsf{cost}_{i\in 0\ldots 7, b\in\{0,1\}})$
    $(\mathbf{k}^{01}, \mathsf{TotalCost}^{01}) \leftarrow \mathsf{Decode}_C^{01}(\mathsf{cost}_{i\in 0\ldots 7, b\in\{0,1\}})$
    if $\mathsf{TotalCost}^{00} < \mathsf{TotalCost}^{01}$ then
        $b = 0$
    else
        $b = 1$
    end if
    $(k_0, k_1, k_2, k_3) \leftarrow \mathbf{k}^{0b}$
    $\mathbf{k}_2 = (k_0, k_1 \oplus k_0, k_3, b)$
    return $\mathbf{k}_2$
end procedure
procedure $\mathsf{Decode}_C^{b_0 b_1}(\mathsf{cost}_{i\in 0\ldots 7, b\in\{0,1\}} \in \mathbb{Z}^{8\times 2})$
    $min_d = +\infty$
    $min_i = 0$
    $\mathsf{TotalCost} = 0$
    for $j = 0 \ldots 3$ do
        $c_0 \leftarrow \mathsf{cost}_{2j,b_0} + \mathsf{cost}_{2j+1,b_1}$
        $c_1 \leftarrow \mathsf{cost}_{2j,1-b_0} + \mathsf{cost}_{2j+1,1-b_1}$
        if $c_0 < c_1$ then
            $k_i \leftarrow 0$
        else
            $k_i \leftarrow 1$
        end if
        $\mathsf{TotalCost} \leftarrow \mathsf{TotalCost} + c_{k_i}$
        if $c_{1-k_i} - c_{k_i} < min_d$ then
            $min_d \leftarrow c_{1-k_i} - c_{k_i}$
            $min_i \leftarrow i$
        end if
    end for
    if $k_0 + k_1 + k_2 + k_3 \bmod 2 = 1$ then
        $k_{min_i} \leftarrow 1 - k_{min_i}$
        $\mathsf{TotalCost} \leftarrow \mathsf{TotalCost} + min_d$
    end if
    $\mathbf{k} = (k_0, k_1, k_2, k_3)$
    return $(\mathbf{k}, \mathsf{TotalCost})$
end procedure
```

We can find $\boldsymbol{\epsilon}, \boldsymbol{\epsilon}_1 \in [-1/2, 1/2]^8, \boldsymbol{\theta} \in \mathbb{Z}^8$ such that

$$\begin{aligned}\left\lfloor \frac{q}{g}\mathbf{v} \right\rceil - \boldsymbol{\sigma}_2 &= \frac{q}{g}\mathbf{v} + \boldsymbol{\epsilon} - \boldsymbol{\sigma}_2 \\ &= \frac{q}{g}\left(\frac{g}{q}\left(\boldsymbol{\sigma}_1 + \frac{q-1}{2}\mathbf{k}_1\mathbf{H}\right) + \boldsymbol{\epsilon} + \boldsymbol{\theta} g\right) + \boldsymbol{\epsilon}_1 - \boldsymbol{\sigma}_2 \\ &= (\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2) + \frac{q-1}{2}\mathbf{k}_1\mathbf{H} + \frac{q}{g}\boldsymbol{\epsilon} + \boldsymbol{\epsilon}_1 + \boldsymbol{\theta} q\end{aligned}$$

Hence, the bias from $\frac{q-1}{2}\mathbf{k}_1\mathbf{H}$ is no larger than $\|\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2\|_{q,2} + \frac{q}{g}\|\boldsymbol{\epsilon}\| + \sqrt{2} \leq \|\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2\|_{q,2} + \sqrt{2}\left(\frac{q}{g} + 1\right)$. If this value is less than the minimal distance $(q-1)/2$, the decoding will be correct, which implies $\mathbf{k}_1 = \mathbf{k}_2$. □

Proposition 1. *AKCN-E8 is* secure. *Specifically, if $\boldsymbol{\sigma}_1$ is subject to uniform distribution over $\mathbb{Z}_q^8$, then $\mathbf{v}$ and $\mathbf{k}_1$ are independent.*

Proof. For arbitrary fixed $\mathbf{k}_1$, $\mathbf{k}_1\mathbf{H} \bmod 2$ is fixed. Since $\boldsymbol{\sigma}_1$ is uniform random, $\boldsymbol{\sigma}_1 + \frac{q}{2}(\mathbf{k}_1\mathbf{H} \bmod 2)$ is uniform random over $\mathbb{Z}_q$. Thus, $\mathbf{v}$ is subject to the distribution $\lfloor \frac{g}{q}\mathbf{u} \rceil \bmod g$, where $\mathbf{u}$ is uniform random over $\mathbb{Z}_q$. Hence, $\mathbf{v}$ is independent of $\mathbf{k}_1$. □

A. Failure Rate Analysis

Now, with respect to the CPA-secure PKE scheme described in Figure 3 with the underlying AKC is replaced with AKCN-E8, we analyze the correctness property by calculating its failure rate.

Denote $\boldsymbol{\epsilon} = \frac{2^t}{q}(\mathbf{a}\mathbf{x}_2 + \mathbf{e}_2) - \lfloor \frac{2^t}{q}(\mathbf{a}\mathbf{x}_2 + \mathbf{e}_2) \rceil$. We have

$$\begin{aligned}\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2 &= \lfloor (\frac{q}{2^t}\mathbf{y}_2)\mathbf{x}_1 \rceil - (\mathbf{y}_1\mathbf{x}_2 + \mathbf{e}_2') \\ &= \boldsymbol{\epsilon'} + (\frac{q}{2^t}\mathbf{y}_2)\mathbf{x}_1 - (\mathbf{y}_1\mathbf{x}_2 + \mathbf{e}_2') \\ &= \frac{q}{2^t}\lfloor \frac{2^t}{q}(\mathbf{a}\mathbf{x}_2 + \mathbf{e}_2) \rceil \mathbf{x}_1 - ((\mathbf{a}\mathbf{x}_1 + \mathbf{e}_1)\mathbf{x}_2 + \mathbf{e}_2') + \boldsymbol{\epsilon'} \\ &= (\mathbf{a}\mathbf{x}_2 + \mathbf{e}_2 - \frac{q}{2^t}\boldsymbol{\epsilon})\mathbf{x}_1 - (\mathbf{a}\mathbf{x}_1\mathbf{x}_2 + \mathbf{e}_1\mathbf{x}_2 + \boldsymbol{e}_2') + \boldsymbol{\epsilon'} \\ &= (\mathbf{e}_2 - \frac{q}{2^t}\boldsymbol{\epsilon})\mathbf{x}_1 - (\mathbf{e}_1\mathbf{x}_2 + \mathbf{e}_2') + \boldsymbol{\epsilon'},\end{aligned}$$

where $\boldsymbol{\epsilon'} \in [-\frac{1}{2}, \frac{1}{2})^4$.

From RLWE assumption, $(\mathbf{a}, \mathbf{a}\mathbf{x}_2 + \mathbf{e}_2)$ is indistinguishable with $(\mathbf{a}, \mathbf{u})$, where $\mathbf{u}$ is subject to the uniform distribution. Then, $\boldsymbol{\epsilon}$ should be closed to $\frac{2^t}{q}\mathbf{u} - \lfloor \frac{2^t}{q}\mathbf{u} \rceil$. Let $\sigma_{q,t}$ be the standard deviation of $\frac{2^t}{q}\mathbf{u} - \lfloor \frac{2^t}{q}\mathbf{u} \rceil$. Then we can calculate the standard deviation of each coefficients of polynomials in $\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2 - \boldsymbol{\epsilon}'$, which we denote it as s. Then we have

$$s^2 = n\sigma^2\left(2\sigma^2 + \frac{q^2}{2^{2t}}\sigma_{q,t}^2\right) + \sigma^2$$

By the Central Limit Theorem, each coefficient of the polynomials in $\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2 - \boldsymbol{\epsilon}'$ is close to a Gaussian distribution. Also note that $\|\boldsymbol{\epsilon}'\|_2 < 1$. Let $\chi^2(8)$ be the chi-square distribution with mean 8. From Theorem 1, the AKCN-E8 scheme is correct with probability

$$\Pr\left[d' \leftarrow \chi^2(8) : \sqrt{d'} \leq \frac{1}{s}\left(\frac{q-1}{2} - \sqrt{2}\left(\frac{q}{g} + 1\right) - 1\right)\right]$$

We provide a script to calculate the concrete failure rate, which is available from http://github.com/AKCN-E8.

B. On the Failure Rate Analysis of NewHope

For the fairness of the comparison, we also provide an estimation of NewHope [1] with the same methodology. We mainly focus on the parameter set for NewHope-1024.

The NewHope-1024 protocol divides the 1024 coefficients of the polynomial $\boldsymbol{\sigma}_1(x)$ and $\boldsymbol{\sigma}_2(x)$ into $1024/4 = 256$ vectors, each of dimension 4. Then they apply the D_4 lattice to extract one bit from each vector. We firstly estimate the failure probability of each bit, and finally take the union bound over the 256 vectors.

Let $\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2 \in \mathbb{Z}_p^4$ be two vectors obtained by the initiator and responder, respectively. If $\|\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2\|_1 < (1 - 2/g) \cdot q - 2$, then the NewHope scheme is correct.

To deal with the 1-norm, we use the same observation as in [1], which asserts that, for any $\mathbf{x} \in \mathbb{R}^4, \|\mathbf{x}\|_1 = \max_{\mathbf{y} \in \{-1,+1\}^4} \langle \mathbf{x}, \mathbf{y} \rangle$. Then, $\|\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2\|_1 = \max_{y \in \{-1,+1\}^4} \langle \boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2, \mathbf{y} \rangle$. By taking union bound on all choices of $\mathbf{y} \in \{-1,+1\}^4$, we only need to analyze for an arbitrary fixed $\mathbf{y} \in \{-1,+1\}^4$, since the distribution of $\langle \boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2, \mathbf{y} \rangle$ is the same for different $\mathbf{y}$. Let s be the standard deviation of $\langle \boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_1, \mathbf{y} \rangle$, then we have $s^2 = 4(2n\sigma^4 + \sigma^2)$. By the Central Limit Theorem, each coordinate of $\boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_1$ is close to a Gaussian distribution. Hence, the failure rate of NewHope is estimated as

$$\Pr\left[d' \leftarrow \mathcal{N}(0,1) : d' \leq \frac{1}{s}\left((1 - 2/g) \cdot q - 2\right)\right]$$

C. On the Tight Upper Bound of the Failure Probability

The recent work [30] suggested using Chernoff-Cramer inequality to give a tight upper bound on the failure rate.

Theorem 2 (Chernoff-Cramer Inequality)**.** *Let D be a distribution over $\mathbb{R}$, and X be the sum of ℓ independent and identically distributed random variables $X_1, X_2, \ldots, X_\ell$. Then, for any $t \in \mathbb{R}, t > 0$, and any $a \in \mathbb{R}$, we have*

$$\Pr[X \geq a] \leq \exp\left(-ta + \ell\ln(\mathbb{E}[e^{tX_i}])\right)$$

Specifically, X_i here is a function of $4 \times 2 = 8$ independent random variables that are subjected to the law of S_η. Hence, the work [30] simulates the distribution by enumerating all possibilities of these 8 random variables, and hence takes $(2\eta + 1)^8 \approx 2^{32}$ operations to compute the distribution of X_i. However, in our case, if following the same analysis, this would take $(2\eta + 1)^{8 \cdot 2} \approx 2^{65}$ operations, which is impossible to compute on an ordinary personal computer. Hence, we use our asymptotic method to give an estimation of the failure probability. We note that the failure probability estimated with our asymptotic method is quite close to the tight bound obtained in [30]. With NewHope-1024 as an instance, its failure probability by our method (resp., by the tight bound in [30]) is 2^{-424} (resp., 2^{-412}).

	$\|K\|$	n	q	η	g	t	pq-sec	err	pk (B)	cipher (B)
NewHope-512-CPA	256	512	12289	8	2^3	14	101	2^{-424}	928	1088
AKCN-E8-512-S-CPA	256	512	12289	14	2^4	11	110	2^{-277}	928	960
AKCN-E8-512-E-CPA	256	512	12289	8	2^4	10	101	2^{-422}	928	896
AKCN-E8-512-C-CPA	256	512	12289	8	2^3	10	101	2^{-252}	928	832
NewHope-512-CCA	256	512	12289	8	2^3	14	101	2^{-424}	928	1120
AKCN-E8-512-S-CCA	256	512	12289	14	2^4	11	110	2^{-277}	928	992
AKCN-E8-512-E-CCA	256	512	12289	8	2^4	10	101	2^{-422}	928	928
AKCN-E8-512-C-CCA	256	512	12289	8	2^3	10	101	2^{-252}	928	864
NewHope-1024-CPA	256	1024	12289	8	2^3	14	233	2^{-424}	1824	2176
NewHope-1024-D2-CPA	512	1024	12289	8	2^3	14	233	2^{-199}	1824	2176
AKCN-E8-1024-S-CPA	512	1024	12289	10	2^4	12	240	2^{-308}	1824	2048
AKCN-E8-1024-E-CPA	512	1024	12289	8	2^4	11	233	2^{-381}	1824	1920
AKCN-E8-1024-C-CPA	512	1024	12289	4	2^3	11	214	2^{-762}	1824	1792
NewHope-1024-CCA	256	1024	12289	8	2^3	14	233	2^{-424}	1824	2208
AKCN-E8-1024-S-CCA	512	1024	12289	10	2^4	12	240	2^{-208}	1824	2080
AKCN-E8-1024-E-CCA	512	1024	12289	8	2^4	11	233	2^{-381}	1824	1952
AKCN-E8-1024-C-CCA	512	1024	12289	4	2^3	11	214	2^{-762}	1824	1824

TABLE I: Parameters for AKCN-E8-KEM and comparison with NewHope-KEM [34]. $|K|$ refers to the size of shared-key $\mathbf{k}_1 = \mathbf{k}_2$ in bits, "pk(B)" refers to the size of $(\mathbf{y}_1, \mathsf{seed})$ in bytes; "cipher(B)" refers to the size of $(\mathbf{y}_2, \mathbf{v})$; "pq-sec" refers to the security of the underlying RLWE problem against the best known quantum attacks, NewHope-1024-D2 refers to the variant of NewHope-1024 that derives 512-bit shared-key with the D_2 code as in NewHope-512.

D. *On the Independence of Errors in Different Positions*

For ease of efficient computation, our asymptotic method of error probability analysis assumes the independence of errors in different positions. It was shown that for the concrete parameters of RLWE-based KEM schemes the independency assumption does not hold [11]. In this work, we show a complementary result by proving the independency of errors when n tends to be infinity.

Suppose $f(x), g(x)$ are two polynomials of degree n, whose coefficients are drawn independently from Gaussian. Let $h(x) = f(x) \cdot g(x) \in \mathbb{R}[x]/(x^n + 1)$. We show that for every two different integers $0 \leq c_1, c_2 < n$, the joint distribution of $(h[c_1], h[c_2])$ will approach to the two-dimensional Gaussian when n tends to infinity. Hence, it is reasonable to assume that the error rates of any two different positions are independent when n is sufficiently large. For representation simplicity, for any polynomial f, let $f[i]$ denote the coefficient of x^i.

Lemma 1. *Suppose $f(x), g(x) \in \mathbb{R}[x]/(x^n + 1)$ are two n-degree polynomials whose coefficients are drawn independently from $\mathcal{N}(0, \sigma^2)$. Let $h(x) = f(x) \cdot g(x) \in \mathbb{R}[x]/(x^n+1)$, where $h(x)$ is represented as an n-degree polynomial. For any two different integers $0 \leq c_1, c_2 < n$, the characteristic function of the two-dimensional random vector $(h[c_1], h[c_2]) \in \mathbb{R}^2$ is*

$$\phi_{c_1,c_2}(t_1, t_2) = \mathbb{E}\left[e^{i(t_1 h[c_1] + t_2 h[c_2])}\right] \tag{1}$$

$$= t_1 \mathbf{f}^T \mathbf{A}_{c_1} \mathbf{g} + t_2 \mathbf{f}^T \mathbf{A}_{c_2} \mathbf{g} \tag{2}$$

$$= \prod_{k=0}^{n-1} \left(1 + \sigma^4(t_1^2 + t_2^2 + 2t_1 t_2\right. \tag{3}$$

$$\left. \cdot \cos(\pi(c_1 - c_2)\frac{2k+1}{n}))\right)^{-\frac{1}{2}} \tag{4}$$

Proof. One can observe that $t_1 h[c_1] + t_2 h[c_2]$ is equal to

$$t_1 \left(\sum_{i+j=c_1} f[i]g[j] - \sum_{i+j=c_1+n} f[i]g[j] \right)$$
$$+ t_2 \left(\sum_{i+j=c_2} f[i]g[j] - \sum_{i+j=c_2+n} f[i]g[j] \right)$$
$$= t_1 \mathbf{f}^T \mathbf{A}_{c_1} \mathbf{g} + t_2 \mathbf{f}^T \mathbf{A}_{c_2} \mathbf{g}. = \mathbf{f}^T (t_1 \mathbf{A}_{c_1} + t_2 \mathbf{A}_{c_2}) \mathbf{g}$$

Where $\mathbf{f} = (f[0], f[1], \ldots, f[n-1])^T$, $\mathbf{g} = (g[0], g[1], \ldots, g[n-1])^T$, and the notations $\mathbf{A}_{c_1}, \mathbf{A}_{c_2}$ are defined by

$$\mathbf{A}_c = \begin{pmatrix} & & 1 & & \\ & \iddots & & & \\ 1 & & & & \\ & & & & -1 \\ & & & \iddots & \\ & & -1 & & \end{pmatrix}$$

The value 1 in the first row is in the c-th column.

As $t_1 \mathbf{A}_{c_1} + t_2 \mathbf{A}_{c_2}$ is symmetric, it can be orthogonally diagonalized as $\mathbf{P}^T \mathbf{\Lambda} \mathbf{P}$, where $\mathbf{P}$ is orthogonal, and $\mathbf{\Lambda}$ is diagonal. Hence, $\phi_{c_1,c_2}(t_1, t_2) = \mathbb{E}[\exp(i(\mathbf{Pf})^T \mathbf{\Lambda}(\mathbf{Pg}))]$. Since $\mathbf{P}$ is orthogonal, it keeps the normal distribution unchanged. Hence, $(\mathbf{Pf})^T \mathbf{\Lambda}(\mathbf{Pg})$ equals the sum of n scaled products of two independent one-dimensional Gaussian.

Suppose $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of $t_1 \mathbf{A}_{c_1} + t_2 \mathbf{A}_{c_2}$, and ϕ is the characteristic function of the product of two independent one-dimensional standard Gaussian. Then we have

$$\phi_{c_1,c_2}(t_1, t_2) = \prod_{k=0}^{n-1} \phi(\sigma^2 \lambda_k) \tag{5}$$

From [37], $\phi(t) = (1 + t^2)^{-1/2}$. For λ_k, we further observe that

$$\begin{aligned}(t_1\mathbf{A}_{c_1}+t_2\mathbf{A}_{c_2})^2 &= (t_1^2+t_2^2)\mathbf{I}+t_1t_2(\mathbf{A}_{c_1}\mathbf{A}_{c_2}+\mathbf{A}_{c_2}\mathbf{A}_{c_1})\\ &= (t_1^2+t_2^2)\mathbf{I}+t_1t_2(\mathbf{G}^{c_2-c_1}+\mathbf{G}^{c_1-c_2}),\end{aligned}$$

where

$$\mathbf{G}=\begin{pmatrix} & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1\\ -1 & & & & \end{pmatrix}$$

The characteristic polynomial of $\mathbf{G}$ is x^n+1. Hence, λ_k satisfies

$$\lambda_k^2=t_1^2+t_2^2+2t_1t_2\cos\left(\pi(c_1-c_2)\frac{2k+1}{n}\right)$$

By taking this into Equation 5, we derive the Equation 4. □

Theorem 3. *For any fixed integers* $0\le c_1,c_2<n$, $c_1\ne c_2$, *when* n *tends to infinity, the distribution of* $\left(\frac{h[c_1]}{\sigma^2\sqrt{n}},\frac{h[c_2]}{\sigma^2\sqrt{n}}\right)$ converges (in distribution) *to the two-dimensional normal distribution* $\mathcal{N}(\mathbf{0},\mathbf{I}_2)$.

Proof. Let $\phi(t_1,t_2)$ denote the characteristic function of the random vector $\left(\frac{h[c_1]}{\sigma^2\sqrt{n}},\frac{h[c_2]}{\sigma^2\sqrt{n}}\right)$. Then, for fixed t_1,t_2,

$$\begin{aligned}\ln(\phi(t_1,t_2)) &= -\frac{1}{2}\sum_{k=0}^{n-1}\ln\Bigg(1+\frac{1}{n}(t_1^2+t_2^2+2t_1t_2 &\quad(6)\\ &\qquad\cdot\cos\left(\pi(c_1-c_2)\frac{2k+1}{n}\right))\Bigg) &\quad(7)\\ &= -\frac{1}{2}\sum_{k=0}^{n-1}\Bigg[\frac{1}{n}(t_1^2+t_2^2+2t_1t_2 &\quad(8)\\ &\qquad\cdot\cos\left(\pi(c_1-c_2)\frac{2k+1}{n}\right))+r_k\Bigg] &\quad(9)\\ &= -\frac{1}{2}\left(t_1^2+t_2^2\right)-\frac{1}{2}\sum_{k=0}^{n-1}r_k, &\quad(10)\end{aligned}$$

where r_k is the Lagrange remainders. So, $|r_k|\le\lambda_k^4/2n^2$. Since $\lambda_k^2\le(|t_1|+|t_2|)^2$, we have $|r_k|\le(|t_1|+|t_2|)^4/2n^2$.

When n tends to infinity, $\phi(t_1,t_2)$ converges pointwise to $\exp(-(t_1^2+t_2^2)/2)$, which is the characteristic function of the two-dimensional normal distribution $\mathcal{N}(\mathbf{0},\mathbf{I}_2)$. From Lévy's convergence theorem, we derive that the random vector $\left(\frac{h[c_1]}{\sigma^2\sqrt{n}},\frac{h[c_2]}{\sigma^2\sqrt{n}}\right)$ *converges in distribution* to the normal distribution $\mathcal{N}(\mathbf{0},\mathbf{I}_2)$. □

V. Parameters and Performance Comparisons

The AKCN-E8-KEM scheme that resulted from the modular and generalized framework described in Section III, with the underlying AKC mechanism replaced with the AKCN-E8 scheme presented in Section IV, works on any hard instantiation of the RLWE problem. But if n is the power of 2, and the prime q satisfies $q \bmod 2n=1$, then number-theoretic transform (NTT) can be used to speed up polynomial multiplication. The performance can be further improved by using the Montgomery arithmetic and AVX2 instruction set [1, 34]. As in [34], the underlying noise distribution is the centered binomial distribution S_η: for some positive integer η, sample $(a_1,\cdots,a_\eta,b_1,\cdots,b_\eta)\leftarrow\{0,1\}^{2\eta}$ and then output $\sum_{i=1}^{\eta}(a_i-b_i)$. For the centered binomial distribution S_η, its standard deviation is $\sigma=\sqrt{\eta/2}$. In NEWHOPE [34], $q=12289$, $n=512$ or $n=1024$, $\eta=8$. For ease of comparison, we use the same CCA transformation and the same values of (q,n) of NewHope [34] for the construction and implementation of AKCN-E8-KEM. We use the same script of NewHope-KEM [34] for concrete security estimation against the underlying RLWE problem by the best known quantum attacks, and omit the details here for presentation simplicity. The reader is referred to [34] for the method and script of concrete security estimation, which is also available from https://newhopecrypto.org/.

The parameters and performance of AKCN-E8-KEM are given in Table I. For both AKCN-E8-512 and AKCN-E8-1024, we present three sets of parameters: "S" stands for higher *security* level, "E" stands for lower *error* probability, and "C" stands for smaller *ciphertext* size. For fairness of comparison, the failure probability for both AKCN-E8 and NewHope are calculated with our asymptotic method. In particular, we introduce the NewHope-1024-D2 variant for fair performance comparison on the same size of shared-key.

A. More Parameters and Comparisons with Kyber and Saber

The standard NTT technique requires that n be power-of-two and $2n|(q-1)$. Recent advances on the variants of NTT [2, 24, 29, 42, 43] allow us to choose the parameters (n,q) in a more flexible way. For example, we can use $q=7681$ and $q=3329$ for AKCN-E8-1024 and AKCN-E8-512. The NTT technique proposed in [24, 29] (resp., in [2]) allows us to use $n=768$ and $q=7681$ (resp., $q=3457$) for AKCN-E8-768. More parameters of AKCN-E8 enabled by the recent advances of NTT techniques are given in Table III (page 11). For presentation simplicity, the parameter sets AKCN-E8-3329-512/1024-CCA (resp., AKCN-E8-7681-768-CCA) are referred to as AKCN-E8-512/768 (resp., AKCN-E8-768) in Tabel IV (page 11).

In Table IV, we make a brief comparison with Kyber and Saber that are the candidate proposals now in the third round of NIST PQC standardization. Note that the key encapsulated by AKCN-E8-512 (resp., 768, 1024) has the size of 256 (resp., 384, 512) bits, which is much more flexible compared to the fixed key size of 256 bits for Kyber and Saber.

B. Implementation and Benchmark

Based on the reference implementation of NewHope-KEM, we provide the implementations of AKCN-E8-KEM, and make performance benchmark comparisons with the NIST reference implementations of NewHope-KEM, Kyber and Saber. The benchmark result for the implementation of AKCN-E8-1024-C-CCA for $q=12289$, together with the performance comparison with NewHope-KEM, is given in Table V (page 11). The benchmark results for the implementations of AKCN-E8-512/768/1024 as specified in Table IV, together with the

	$\lvert K\rvert$	n	q	η	g	t	pq-sec	err	pk (B)	cipher (B)
AKCN-E8-7681-512-CPA	256	512	7681	6	2^3	3	104	2^{-204}	864	832
AKCN-E8-7681-512-CCA	256	512	7681	6	2^3	3	104	2^{-204}	864	864
AKCN-E8-7681-640-CPA	320	640	7681	6	2^3	3	137	2^{-159}	1072	1040
AKCN-E8-7681-640-CCA	320	640	7681	6	2^3	3	137	2^{-159}	1072	1072
AKCN-E8-7681-768-CPA	384	768	7681	4	2^3	3	161	2^{-245}	1280	1248
AKCN-E8-7681-768-CCA	384	768	7681	4	2^3	3	161	2^{-245}	1280	1280
AKCN-E8-1024-CPA-Recom	512	1024	7681	4	2^4	3	227	2^{-303}	1696	1792
AKCN-E8-1024-CPA-Option-S	512	1024	7681	6	2^4	2	239	2^{-267}	1696	1960
AKCN-E8-1024-CPA-Option-C	512	1024	7681	2	2^3	3	208	2^{-471}	1696	1664
AKCN-E8-1024-CCA-Recom	512	1024	7681	4	2^4	3	227	2^{-303}	1696	1824
AKCN-E8-1024-CPA-Option-S	512	1024	7681	6	2^4	2	239	2^{-267}	1696	1992
AKCN-E8-1024-CCA-Option-C	512	1024	7681	2	2^3	3	208	2^{-471}	1696	1696

TABLE II: Recommended parameters for AKCN-E8-7681. "Recom" (resp., "Option") stands for "Recommended" (resp., "Optional"). We recommend to use the same $q = 7681$ and $\eta = 4$ for all the three sets of parameters.

	$\lvert K\rvert$	n	q	η	g	t	pq-sec	err	pk (B)	cipher (B)
AKCN-E8-3329-512-CPA	256	512	3329	3	2^3	10	107	2^{-193}	800	832
AKCN-E8-3329-512-CCA	256	512	3329	3	2^3	10	107	2^{-193}	800	864
AKCN-E8-7681-768-CPA	384	768	7681	4	2^3	9	161	2^{-245}	1280	1248
AKCN-E8-7681-768-CCA	384	768	7681	4	2^3	9	161	2^{-245}	1280	1280
AKCN-E8-3329-1024-CPA	512	1024	3329	2	2^3	10	230	2^{-178}	1568	1664
AKCN-E8-3329-1024-CCA	512	1024	3329	2	2^3	10	230	2^{-178}	1568	1696

TABLE III: More parameters for AKCN-E8

	$\lvert K\rvert$	q	pq-sec	err	pk (B)	cipher (B)
AKCN-E8-512	256	3329	107	2^{-193}	800	864
Kyber-512	256	3329	107	2^{-139}	800	768
LightSaber	256	2^{13}	107	2^{-120}	672	736
AKCN-E8-768	384	7681	161	2^{-245}	1280	1280
Kyber-768	256	3329	164	2^{-164}	1184	1088
Saber	256	2^{13}	172	2^{-136}	992	1088
AKCN-E8-1024	512	3329	230	2^{-178}	1568	1696
Kyber-1024	256	3329	230	2^{-174}	1568	1568
FireSaber	256	2^{13}	236	2^{-165}	1312	1472

TABLE IV: Comparisons with Kyber and Saber

performance comparisons with Kyber and Saber, are summarized in Table VI (page 12). All the source codes are available from http://github.com/AKCN-E8.

We implement the algorithms on macOS version 11.0, clang version 12.0.0.31.1. We run the benchmark on 8-Core Intel Core i7-9700K processor clocked at 3.6 GHz with Hyper-Threading off. The code is compiled with the option -O3 -fomit-frame-pointer -march=native. We run key generation, encryption and decryption each for 10000 times. The reported time and CPU cycles are the medians of the cycle counts.

	AKCN-E8-1024-CCA		NewHope-1024-CCA	
	Time(us)	Cycle	Time(us)	Cycle
Gen	59	213371	69	249161
Enc	89	321427	101	362497
Dec	117	423812	117	421997

TABLE V: Benchmark of AKCN-E8 for $q = 12289$

Appendix

The standard NTT technique requires that $q \mod 2n = 1$. Recent advances on the variants of NTT [2, 29, 42, 43] allow us to choose the module q in a more flexible way. For example, we can use $q = 7681$ and $q = 3329$ for AKCN-E8-1024 and AKCN-E8-512. The NTT technique proposed in [29] (resp., in [2]) allows us to use $q = 7681$ (resp., $q = 3457$) for AKCN-E8-768. A variant of the NTT technique [29] also allows us to use $q = 7681$ for AKCN-E8-640. More parameters of AKCN-E8 enabled by the recent advances of NTT techniques are given in Table II and Table III. We may prefer to the AKCN-E8-7681 parameter sets, as they share the same module $q = 7681$ for AKCN-E8-512, AKCN-E8-640, AKCN-E8-768 and AKCN-E8-1024.

References

[1] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum Key Exchange — A New Hope. *25th USENIX Security Symposium (USENIX Security 16)*, pages 327-343. Winner of the 2016 Internet Defense Prize (https://internetdefenseprize.org/)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2022.3148586, IEEE Transactions on Information Theory

	AKCN-E8			Kyber			Saber		
n	512	768	1024	512	768	1024	512	768	1024
Gen	121380	260126	214018	123340	207838	326526	71959	124009	198180
Enc	145342	269388	258164	150350	236516	347980	82295	148283	222560
Dec	195256	366892	258392	175784	270780	391586	83581	149155	243936

TABLE VI: Benchmarks of AKCN-E8, Kyber and Saber

[2] E. Alkim, Y. A. Bilgin, and M. Cenk. Compact and Simple RLWE Based Key Encapsulation Mechanism. *LATINCRYPT* 2019: 237-256.

[3] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. *CRYPTO 2009*: 595-618.

[4] J. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila. Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE. *ACM CCS 2016*: 1006-1018.

[5] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. Schanck, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. *EuroS&P* 2018: 353-367.

[6] H. Cohn, A. Kumar, S. D. Miller, D. Radchenko, M. Viazovska. The Sphere Packing Problem in Dimension 24. *Annuals of Mathematics*, 185 (3): 1017-1033, 2017.

[7] J. Conway and N. Sloane. Fast quantizing and decoding algorithm for lattice quantizers and codes. *IEEE Transactions on Information Theory*, 28 (2): 227-232, 1982.

[8] J. Conway and N. Sloane. Sphere Packings, Lattices, and Groups. Springer- Verlag, New York, 1993.

[9] R. Cramer, I. Damgärd and B. Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. *CRYPTO* 1994: 174-187.

[10] J. D'Anvers, A. Karmakar, S. Roy and F. Vercauteren. Saber: Module-LWR Based Key Exchange, CPA-Secure Encryption and CCA-Secure KEM. *AFRICACRYPT* 2018: 282-305.

[11] J. D'Anvers, F. Vercauteren and I. Verbauwhede. The Impact of Error Dependencies on Ring/Mod-LWE/LWR Based Schemes. *PQCrypto* 2019: 103-115.

[12] A. W. Dent. A DesignerÂ¡Â¯s Guide to KEMs. *Cryptology ePrint Archive*, Report 2002/174, 2002.

[13] L. Ducas and A. Durmus. Ring-LWE in Polynomial Rings. *PKC 2012*: 34-51.

[14] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. *CRYPTO* 1986: 186-194.

[15] E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* Volume 83, Issue 1, pages 24-32, 1999.

[16] E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. *Journal of Cryptology*, Volume 26, Issue 1, pp. 8-101, 2013.

[17] Q. Guo, T. Johansson and J. Yang. A Novel CCA Attack Using Decryption Errors Against LAC. *ASIACRYPT* (1) 2019: 82-111.

[18] D. Hofheinz, K. Hövelmanns, and E. Kiltz. A Modular Analysis of the Fujisaki-Okamoto Transformation. *TCC* (1) 2017: 341-371.

[19] K. Hövelmanns, E. Kiltz, S. Schäge, and D. Unruh. Generic Authenticated Key Exchange in the Quantum Random Oracle Model. *Cryptology ePrint Archive*, Report 2018/928.

[20] S. Jaques, M. Naehrig, M. Roetteler, and F. Virdia. Implementing Grover Oracles for Quantum Key Search on AES and LowMC. *Eurocrypt* (2) 2020: 280-310.

[21] H. Jiang, Z. Zhang, and Z. Ma. Tighter Security Proofs for Generic Key Encapsulation Mechanism in the Quantum Random Oracle Model. *PQCrypto* 2019: 227-248.

[22] Z. Jin, and Y. Zhao. Optimal Key Consensus in Presence of Noise. CoRR, abs/1611.06150 (2016) https://arxiv.org/abs/1611.06150

[23] Z. Jin, and Y. Zhao. Generic and Practical Key Establishment from Lattice. ACNS 2019: 302-322.

[24] Z. Liang, S. Shen, Y, Shi, D. Sun, C. Zhang, G. Zhang, Y. Zhao and Z. Zhao. Number Theoretic Transform: Generalization, Optimization, Concrete Analysis and Applications. Inscrypt 2020.

[25] R. Lindner and C. Peikert. Better Key Sizes (and Attacks) for LWE-Based Encryption. *CT-RSA 2011*: 319-339.

[26] X. Lu, Y. Liu, D. Jia, H. Xue, J. He, Z. Zhang, Z. Liu, H. Yang, B. Li, K. Wang. Supporting documentation: LAC. Technical report, National Institute of Standards and Technology.

[27] V. Lyubashevsky, C. Peikert, and O. Regev. On Ideal Lattices and Learning with Errors over Rings. *EUROCRYPT 2010*: 1-23.

[28] V. Lyubashevsky, C. Peikert, and O. Regev. A Toolkit for Ring-LWE Cryptography. *EUROCRYPT 2013*: 35-54.

[29] V. Lyubashevsky and G. Seiler. NTTRU: Truly Fast NTRU Using NTT. *CHES* 2019: 180-201.

[30] Thomas Plantard and Arnaud Sipasseuth and Willy Susilo and Vincent Zucca Tight bound on NewHope failure probability, IACR Cryptol. ePrint Arch. 2019: 1451.

[31] NIST. Post-Quantum Cryptography Standardization. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization

[32] C. Peikert, O. Regev and N. Stephens-Davidowitz. Pseudorandomness of Ring-LWE for Any Ring and Modulus. *STOC* 2017: 461-473.

[33] A.V. Poppelen, Cryptographic Decoding of the Leech Lattice. *Cryptology ePrint Archive*, Report 2016/1050,

2016.

[34] T. Pöppelmann, E. Alkim, R. Avanzi, J. Bos, L. Ducas, A. Piedra, P. Schwabe, D. Stebila, M. Albrecht, E. Orsini, V. Osheter, K. Paterson, G. Peer, and N. Smart. Supporting documentation: Newhope. Technical report, National Institute of Standards and Technology. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/round-2-submissions

[35] T. Pöppelmann and T. Güneysu. Towards Practical Lattice-Based Public-Key Encryption on Reconfigurable Hardware. *SAC 2013*: 68-85.

[36] O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. *Journal of the ACM (JACM)*, Volume 56, Issue 6, pages 34, 2009.

[37] M.K. Simon. Probability Distributions Involving Gaussian Random Variables : A Handbook for Engineers and Scientists. Springer, 2012.

[38] E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3, RFC 8446, 2018.

[39] E. E. Targhi and D. Unruh. Post-Quantum Security of the Fujisaki-Okamoto and OAEP Transforms. *TCC* 2016-B: 192-216.

[40] A. Vardy, and Y. Be'ery. Maximum Likelihood Decoding of the Leech Lattice. *IEEE Transactions on Information Theory*, 39(4):1435-1444, 1993.

[41] M. S. Viazovska. The Sphere Packing Problem in Dimension 8. *Annuals of Mathematics*, 185(3): 991-1015, 2017.

[42] S. Zhou, H. Xue, D. Zhang, K. Wang, X. Lu, B. Li, and J. He. Preprocess-then-NTT Techniques and Its Applications to Kyber and NewHope. *Inscrypt* 2018: 117-137.

[43] Y. Zhu, Y. Pan and Z. Liu. When NTT Meets Karatsuba: Preprocess-then-NTT Technique Revisited. *Cryptology ePrint Archive*, 2019/1079.

Zhengzhong Jin is a fifth year Ph.D. student at Johns Hopkins University. His research interest lies in cryptography, coding theory, and related fields in theoretical computer science. Currently, he is working on proof systems, multi-party computation, and concretely efficient cryptosystems.

Shiyu Shen is a Ph.D. candidate at School of Computer Science, Fudan university. Her research interests include lattice-based cryptography, applied cryptography and cryptographic engineering.

Yunlei Zhao received the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2004. He joined the Hewlett-Packard European Research Center, Bristol, U.K., as a Post-Doctoral Researcher, in 2004. Since 2005, he has been with Fudan University, where he is currently a Professor with the School of Computer Science. His research interests are the theory and applications of cryptography.