

Compact Lattice-Coded (Multi-Recipient) Kyber without CLT Independence Assumption

Shuiyin Liu¹[0000–0002–3762–8550] and Amin Sakzad²[0000–0003–4569–3384]

¹ Holmes Institute, Melbourne, VIC 3000, Australia
SLiu@holmes.edu.au

² Monash University, Melbourne, VIC 3800, Australia
Amin.Sakzad@monash.edu

Abstract. This work presents a joint design of encoding and encryption procedures for public key encryptions (PKEs) and key encapsulation mechanism (KEMs) such as Kyber, without relying on the assumption of independent decoding noise components, achieving reductions in both communication overhead (CER) and decryption failure rate (DFR). Our design features two techniques: ciphertext packing and lattice packing. First, we extend the Peikert-Vaikuntanathan-Waters (PVW) method to Kyber: ℓ plaintexts are packed into a single ciphertext. This scheme is referred to as P_ℓ -Kyber. We prove that the P_ℓ -Kyber is IND-CCA secure under the M-LWE hardness assumption. We show that the decryption decoding noise entries across the ℓ plaintexts (also known as layers) are mutually independent. Second, we propose a cross-layer lattice encoding scheme for the P_ℓ -Kyber, where every ℓ cross-layer information symbols are encoded to a lattice point. This way we obtain a *coded* P_ℓ -Kyber, where the decoding noise entries for each lattice point are mutually independent. Therefore, the DFR analysis does not require the assumption of independence among the decryption decoding noise entries. Both DFR and CER are greatly decreased thanks to ciphertext packing and lattice packing. We demonstrate that with $\ell = 24$ and Leech lattice encoder, the proposed coded P_ℓ -KYBER1024 achieves $\text{DFR} < 2^{-281}$ and $\text{CER} = 4.6$, i.e., a decrease of CER by 90% compared to KYBER1024. Additionally, for a fixed plaintext size matching that of standard Kyber (256 bits), we introduce a truncated variant of P_ℓ -Kyber that deterministically removes ciphertext components carrying surplus information bits. Using $\ell = 8$ and E8 lattice encoder, we show that the proposed truncated coded P_ℓ -KYBER1024 achieves a 10.2% reduction in CER and improves DFR by a factor of 2^{30} relative to KYBER1024. Finally, we demonstrate that constructing a multi-recipient PKE and a multi-recipient KEM (mKEM) using the proposed truncated coded P_ℓ -KYBER1024 results in a 20% reduction in bandwidth consumption compared to the existing schemes.

Keywords: Module leaning with errors · public key encryption · Ciphertext packing · Lattice packing · Key Encapsulation Mechanisms · Ciphertext expansion · Multi-Recipient

1 Introduction

In August 2024, National Institute of Standards and Technology (NIST) has published the final post-quantum cryptography standards for digital-signature, encryption, and key-encapsulation mechanisms (KEM). CRYSTALS-Kyber is the only post-quantum KEM standardised by NIST [25]. In February 2024, Apple has announced that its iMessage is going to use Kyber [3], where sender devices generate post-quantum encryption keys using the receiver’s public keys. Kyber is a lattice-based cryptographic algorithm built upon the module-learning with errors (M-LWE) problem. Unlike traditional KEMs like Elliptic-curve Diffie–Hellman (ECDH), the Kyber algorithm results in much larger ciphertext size (e.g., up to 49 times larger), which necessitates more storage, increased memory usage, and greater demand for network bandwidth. Later, Facebook reported an increase of about 40% in CPU cycles after implementing Kyber, compared with its current ECDH [32]. For the upcoming large-scale deployment phase, it is crucial to enhance Kyber in order to reduce its storage, memory usage, and communication bandwidth. In early 2025, NIST announced adaptation of HQC as another KEM built upon code-based cryptographic assumptions.

Kyber’s encryption and decryption processes can be viewed as a noisy communication channel with binary Pulse Amplitude Modulation (2-PAM) [23]. Recent coding schemes aim to encode more bits, reducing the ciphertext expansion rate (CER). For instance, [19] replaces 2-PAM with a Leech lattice constellation, achieving a 32.6% CER reduction. [23] proposes a 5-PAM Q-ary BCH encoding, cutting CER by 45.6%. To the best of our knowledge, [20] achieves the best results so far. That is a 54% CER reduction by transforming Kyber’s processes into a Gaussian channel, encoding a 638-bit secret in a single ciphertext. These methods rely on an independence assumption for decryption noise, using the central limit theorem to model noise as Gaussian. While valid for higher dimensions, this assumption may underestimate the decryption failure rate (DFR), as the noise entries are actually dependent.

Kyber uses a lossy compression (quantization) function to reduce ciphertext size, increasing decoding noise. Efforts to reduce quantization noise focus on minimizing channel noise. [2] applied D_4 -lattice quantization to Ring-LWE (R-LWE), and [31] extended this to M-LWE with E_8 -lattice quantization. [18] proposed a lattice quantization framework for Kyber, reducing the CER by 36.47% and DFR by a factor of 2^{99} . [20] showed that Lloyd-Max quantization minimizes mean squared error (MMSE) for M-LWE samples. However, the DFR analysis in [18][20] still assumes independence in decoding noise entries.

The independence assumption mentioned above is closely related to the Central Limit Theory (CLT): the dominant decoding noise components are assumed to be i.i.d. Gaussian random variables. The independent/CLT assumption also appears in the wider literature on (R/M-) LWE based cryptosystem and fully homomorphic encryption (FHE). In R-LWE, the independent assumption was applied to the decoding noise entries of NewHope-Simple [15]. In LWE, the Gaussian approximation was applied to the decoding noise entries of FrodoKEM [22]. The CLT assumption was also used in the homomorphic encryption (HE)

schemes [8][24]. An open question is how to create a (R/M-) LWE based *coded* cryptosystem that does not depend on the independence assumption on the decoding noise entries.

Apart from the coding approach, an alternative way of reducing the CER is ciphertext packing [26] (also well known as PVW packing), which was originally proposed for LWE. The authors show that a single LWE ciphertext vector (i.e., the first part of the ciphertext) can be securely reused to encrypt multiple ciphertexts (i.e., 1-bit messages) by employing various secret-key vectors. The resulting cryptosystem is much more efficient than Regev’s scheme [29], since the CER can be made as small as a constant by packing many plaintexts. This method has been used to construct LWE based HE schemes [7][6] and KEM scheme [1]. The downside of ciphertext packing is the increased DFR, due to a union bound probability of decryption failure for each plaintext. To the best of the authors’ knowledge, the ciphertext packing has not been introduced to M-LWE based KEMs like Kyber. It would be interesting to see how the ciphertext packing method affects the DFR analysis, CER, and security level of Kyber.

Although Kyber is originally specified to encapsulate a 256-bit secret, many key encapsulation mechanisms—particularly those used in contexts involving Forward Secrecy (FS) and Key Rotation—frequently exchange raw shared secrets exceeding 256 bits. For instance, TLS 1.3 employs P-384 for FS, yielding a 384-bit raw secret prior to key derivation [30]. Apple’s PQ3 protocol employs a hybrid key exchange scheme that combines ECC and Kyber, producing a concatenated raw shared secret of approximately 512 bits prior to input to the key derivation function (KDF) [3]. From a multi-key derivation perspective, many protocols require derivation of several symmetric keys (e.g., AES key, HMAC key, IV) from a single exchange. Even when only a 256-bit AES key is ultimately used, a larger raw secret enhances KDF resilience and entropy distribution. From a theoretical perspective, [23] demonstrated that encoding across four standard Kyber ciphertexts can yield a 2214-bit secret. More recently, [20] showed that it is feasible to embed two AES keys within a single standard Kyber ciphertext. These findings suggest the potential for constructing more compact M-LWE-based KEMs than the current Kyber scheme. Accordingly, we argue that extending Kyber to support the encapsulation of larger secrets is both practically meaningful and theoretically significant for modern cryptographic applications.

The standard single-recipient KEM can be generalized to support multiple recipients (mKEM) by accepting multiple public keys as input [33]. This is useful in scenarios where the same session key \mathbf{m} needs to be securely shared with a group of recipients. A key advantage of mKEM is its reduced ciphertext overhead compared to the naive approach of performing individual encryptions for each recipient. In [17], Kyber-based mPKE and mKEM schemes were introduced, wherein the session key \mathbf{m} is encapsulated using multiple public keys and combined into a single ciphertext. This approach achieves a significant reduction in ciphertext size relative to the naive method. Moreover, any further reduction in the ciphertext size of the underlying Kyber scheme directly translates to a corresponding reduction in the ciphertext size of the Kyber-based mKEM.

Our main contribution is to develop a M-LWE based KEM with a very low CER (e.g., $\text{CER} < 5$), where its DFR analysis does not depend on the assumption of independence among the decryption decoding noise entries. Our design leverages two techniques: ciphertext packing and lattice packing. The former greatly reduces the CER, while the latter is effective at decreasing the DFR (i.e., compensating the drawback of ciphertext packing). Below we summarize the means that we achieve this:

- We first propose a packed version of Kyber: ℓ plaintexts are packed into a single ciphertext. This scheme is referred to as P_ℓ -Kyber. We prove that the P_ℓ -Kyber is IND-CCA secure under the M-LWE hardness assumption. We show that the decryption decoding noise entries across the ℓ plaintexts (also known as layers) are mutually independent. We also propose a cross-layer lattice encoding scheme for the P_ℓ -Kyber, where every ℓ cross-layer information symbols are encoded to a lattice point. This way we obtain a *coded* P_ℓ -Kyber, which takes the advantages of both ciphertext packing and lattice packing. An upper bound on the DFR of coded P_ℓ -Kyber is derived, which can be verified numerically. We demonstrate that with $\ell = 24$ and Leech lattice encoder, the proposed coded P_ℓ -KYBER1024 achieves $\text{DFR} \leq 2^{-281}$ and $\text{CER} = 4.6$ (see Table 5).
- Secondly, for a fixed plaintext size equivalent to that of standard Kyber (256 bits), we propose a truncated variant of P_ℓ -Kyber that deterministically eliminates ciphertext components conveying redundant information bits. Employing $\ell = 8$ in conjunction with the E8 lattice encoder, the proposed truncated coded P_ℓ -KYBER1024 achieves a 10.2% reduction in the CER and yields a DFR improvement by a factor of 2^{30} compared to KYBER1024. Additionally, we finally demonstrate that implementing a multi-recipient KEM (mKEM) based on the proposed truncated coded P_ℓ -KYBER1024 achieves a 20% reduction in bandwidth usage compared to existing mKEM schemes.

A summary of our main results is provided in Table 1 for convenience.

2 Preliminaries

In this section, we set the notations, provide the definitions and background on coding techniques. We further provide Kyber algorithms and identify the gaps in analysis regarding the independence assumptions used in central limit theorem (CLT) in various prior works.

2.1 Notation and Definitions

Rings: Let R_q denote the polynomial ring $\mathbb{Z}_q[X]/(X^n + 1)$, where $n = 256$ and $q = 3329$ in this setting. Elements of R_q are represented by regular font letters, while vectors of coefficients in R_q are denoted by bold lowercase letters. Matrices and vectors are indicated by bold uppercase and lowercase letters, respectively. The

Table 1. Variants of KYBER1024 for encrypting τ AES Keys by packing ℓ ciphertexts. N : total plaintext size (in bytes), M : total ciphertext size (in bytes), δ : DFR, ρ : CER

DFR Analysis	CLT			Numerical		
Scheme	[19]	[23]	[20]	[25]	This work	
Encoder	Lattice ²	Q-BCH	Binary-BCH	Uncoded	Uncoded	Lattice ²
MMSE ¹ Quantizer	No	No	Yes	No	Yes	Yes
$\tau = 1$ (1 AES key)	$N = 32$ $M = 1184$ $\ell = 1$ $\delta = 2^{-213}$ $\rho = 37$	$N = 58$ $M = 1568$ $\ell = 1$ $\delta < 2^{-174}$ $\rho = 26.6$	-	$N = 32$ $M = 1568$ $\ell = 1$ $\delta = 2^{-174}$ $\rho = 49$	$N = 32$ $M = 1568$ $\ell = 1$ $\delta = 2^{-190}$ $\rho = 49$	$N = 32$ $M = 1408$ $\ell = 8$ $\delta = 2^{-204}$ $\rho = 44$
$\tau = 2$ (2 AES keys)	-	-	$N = 79$ $M = 1792$ $\ell = 1$ $\delta = 2^{-174}$ $\rho = 22.5$	-	$N = 64$ $M = 1728$ $\ell = 2$ $\delta = 2^{-189}$ $\rho = 27$	$N = 64$ $M = 1536$ $\ell = 8$ $\delta = 2^{-203}$ $\rho = 24$
$\tau = 8$ (8 AES keys)	-	$N = 276$ $M = 6272$ $\ell = 4$ $\delta < 2^{-174}$ $\rho = 22.7$	-	-	$N = 256$ $M = 2688$ $\ell = 8$ $\delta = 2^{-187}$ $\rho = 10.5$	$N = 256$ $M = 2688$ $\ell = 8$ $\delta = 2^{-336}$ $\rho = 10.5$

¹ MMSE quantization is defined in Definition 2.

² Lattice coding principles are detailed in Section 2.2. The encoding scheme for 1–2 AES keys is described in Section 5, while those for 8–36 AES keys are presented in Section 4.

transpose of a vector \mathbf{v} or a matrix \mathbf{A} is represented as \mathbf{v}^T or \mathbf{A}^T , respectively. By default, vectors are treated as column vectors.

Sampling and Distribution: For a set \mathcal{S} , we use the notation $s \leftarrow \mathcal{S}$ to indicate that s is chosen uniformly at random from \mathcal{S} . If \mathcal{S} represents a probability distribution, this means s is chosen according to that distribution. This notation is extended coefficient-wise to a polynomial $f(x) \in R_q$ or a vector of such polynomials. Let x be a bit string and S be a distribution that takes x as input. We express $y \sim S := \text{Sam}(x)$ to mean that the output y generated by the distribution S using input x can be extended to any desired length. We define $\beta_\eta = B(2\eta, 0.5) - \eta$ as the central binomial distribution over \mathbb{Z} . The Cartesian product of two sets A and B is represented as $A \times B$. We denote $A \times A$ as A^2 .

Compression and Quantization: Given $x \in \mathbb{R}$, the notation $\lceil x \rceil$ refers to rounding x to the nearest integer, with ties rounded up. The operations $\lfloor x \rfloor$ and $\lceil x \rceil$ denote rounding x down and up, respectively. Now, considering $x \in \mathbb{Z}_q$ and $d \in \mathbb{Z}$ such that $2^d < q$, Kyber compression and decompression functions are [25]:

$$\begin{aligned}
 x' &= \text{Compress}_q(x, d) = \lceil (2^d/q) \cdot x \rceil \pmod{2^d}, \\
 \hat{x} &= \text{Decompress}_q(x', d) = \lceil (q/2^d) \cdot x' \rceil \in \mathcal{C}.
 \end{aligned} \tag{1}$$

Kyber compression and decompression operations can be interpreted as a mapping from a large set \mathbb{Z}_q to a smaller set \mathcal{C} with $|\mathcal{C}| = 2^d < q$. In the literature of signal processing, this mapping is generally known as *quantization*.

Definition 1 (Scalar Quantization). *Given a random variable $x \in \mathbb{Z}_q$ and an integer $L > 0$, a scalar quantization Q_L divides the support of x into L subsets R_1, \dots, R_L , referred to as quantization regions $T_L = \bigcup_{i=1}^L R_i$. Each region R_j is associated with a quantizer $\alpha_j \in \mathcal{C}_L$. When x lies within the region R_j , the quantization Q_L maps x to the point $\hat{x} = \alpha_j$. Q_L can be viewed as a function:*

$$Q_L : \mathbb{Z}_q \rightarrow \mathcal{C}_L, \quad (2)$$

where $Q_L(x, \mathcal{C}_L, T_L) := \hat{x}$ can be uniquely represented by its index in \mathcal{C}_L , denoted as $\text{Index}_L(\hat{x})$, i.e., $\mathcal{C}_L(\text{Index}_L(\hat{x})) = \hat{x}$. The communication cost of transmitting \hat{x} reduces to $\log_2(L)$ bits.

For consistency, with $L = 2^d$, the Kyber quantization in (1) is redefined as

$$\begin{aligned} \hat{x} &= Q_{\text{Kyber}, 2^d}(x) = \text{Decompress}_q(\text{Compress}_q(x, d), d) \\ x' &= \text{Index}_{2^d}(\hat{x}) = \text{Compress}_q(x, d). \end{aligned} \quad (3)$$

Definition 2 (MMSE Quantization). *The optimal quantization should minimize the mean squared quantization error (MMSE):*

$$(\mathcal{C}_L, T_L) = \arg \min_{\mathcal{C}'_L \in \mathbb{R}^n, T'_L \subset \mathbb{R}^n} \mathbb{E}(\|\mathbf{x} - Q_L(\mathbf{x}, \mathcal{C}'_L, T'_L)\|^2). \quad (4)$$

For simplicity of notation, we define the MMSE quantization as

$$\hat{\mathbf{x}} = Q_{\text{MMSE}, L}(\mathbf{x}). \quad (5)$$

The MMSE scalar quantization is the Lloyd-Max quantization [21].

Definition 3 (Moment Generating Function). *Let $X \leftarrow D$ be a random variable. For $\theta \in \mathbb{R}$, the moment generating function (MGF) of X is denoted by*

$$M_X(\theta) = \mathbb{E}(\exp(\theta X)). \quad (6)$$

Definition 4 (Algebraic Expression of a Column Ring Vector). *A column ring vector $\mathbf{v} \in R_q^\ell$ is defined as:*

$$\mathbf{v} = [v_0(x), v_1(x), \dots, v_{\ell-1}(x)]^T, \text{ where } v_i(x) = \sum_{j=0}^{n-1} v_{i,j} x^j \in R_q, \quad v_{i,j} \in \mathbb{Z}_q.$$

Define the mapping function $\phi : R_q^\ell \rightarrow \mathbb{Z}_q^{\ell \times n}$:

$$\phi(\mathbf{v}) = \begin{bmatrix} v_{0,0} & v_{0,1} & \cdots & v_{0,n-1} \\ v_{1,0} & v_{1,1} & \cdots & v_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{\ell-1,0} & v_{\ell-1,1} & \cdots & v_{\ell-1,n-1} \end{bmatrix}.$$

The function ϕ extracts the coefficients of each ring element $v_i(x)$ and arranges them as the i -th row of $\phi(\mathbf{v}) \in \mathbb{Z}_q^{\ell \times n}$. The inverse mapping $\phi^{-1} : \mathbb{Z}_q^{\ell \times n} \rightarrow R_q^\ell$ reconstructs the ring vector from its coefficient matrix: $\mathbf{v} = \phi^{-1}(\phi(\mathbf{v}))$.

2.2 Lattice Code, Encoder, and Decoder

Definition 5 (Lattice). An ℓ -dimensional lattice Λ is a discrete additive subgroup of \mathbb{R}^M with $M \geq \ell$. Given ℓ linearly independent column vectors $\mathbf{b}_1, \dots, \mathbf{b}_\ell \in \mathbb{R}^M$, the lattice generated by these vectors is defined as:

$$\Lambda = \mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^{\ell} z_i \mathbf{b}_i \mid z_i \in \mathbb{Z} \right\},$$

where $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_\ell]$ is referred to as a generator matrix of Λ .

Definition 6 (Lattice Code). A lattice code $\mathcal{C}(\Lambda, \mathcal{P})$ is the finite collection of points in Λ that fall within the bounded set \mathcal{P} :

$$\mathcal{C}(\Lambda, \mathcal{P}) = \Lambda \cap \mathcal{P}.$$

If $\mathcal{P} = \mathbb{Z}_p^\ell$, the code $\mathcal{C}(\Lambda, \mathbb{Z}_p^\ell)$ is said to be generated from hypercube shaping (HS).

Definition 7 (CVP Decoder). Given $\mathbf{y} \in \mathbb{R}^\ell$, the Closest Vector Problem (CVP) decoder returns the nearest lattice vector to \mathbf{y} within the lattice $\mathcal{L}(\mathbf{B})$:

$$\mathbf{x} = \text{CVP}(\mathbf{y}, \mathcal{L}(\mathbf{B})) = \arg \min_{\mathbf{x}' \in \mathcal{L}(\mathbf{B})} \|\mathbf{x}' - \mathbf{y}\|.$$

Definition 8 (HS Encoder [22]). Let $\mathbf{B} = \mathbf{U} \cdot \text{diag}(\pi_1, \dots, \pi_\ell) \cdot \mathbf{U}'$ be the Smith Normal Form (SNF) factorization of a lattice basis \mathbf{B} , where \mathbf{U} and \mathbf{U}' are unimodular matrices in $\mathbb{Z}^{\ell \times \ell}$. Let the message space be

$$\mathcal{M}_{p,\ell} = \{0, 1, \dots, p/\pi_1 - 1\} \times \dots \times \{0, 1, \dots, p/\pi_\ell - 1\}, \quad (7)$$

where $p > 0$ is a common multiple of π_1, \dots, π_ℓ . Given an input $\mathbf{m} \in \mathcal{M}_{p,\ell}$ and $\hat{\mathbf{B}} = \mathbf{U} \cdot \text{diag}(\pi_1, \dots, \pi_\ell)$, a HS encoder produces a codeword $\mathbf{x} \in \mathcal{C}(\mathcal{L}(\mathbf{B}), \mathbb{Z}_p^\ell)$:

$$\mathbf{x} = \hat{\mathbf{B}} \mathbf{m} \bmod p, \quad (8)$$

Definition 9 (HS CVP Decoder [22]). Given a lattice $\mathcal{L}(\mathbf{B})$ in \mathbb{R}^M and an input vector $\mathbf{y} \in \mathbb{R}^M$, the HS CVP decoder outputs an estimated message $\hat{\mathbf{m}} = [\hat{m}_1, \dots, \hat{m}_\ell]^T \in \mathcal{M}_{p,\ell}$:

$$\begin{aligned} \hat{\mathbf{m}} &= \text{CVP}_{\text{HS}}(\mathbf{y}, \mathcal{L}(\mathbf{B})) \\ &= \hat{\mathbf{B}}^{-1} \cdot \text{CVP}(\mathbf{y}, \mathcal{L}(\mathbf{B})) \bmod (p/\pi_1, \dots, p/\pi_\ell), \end{aligned} \quad (9)$$

where $\hat{m}_i = (\hat{\mathbf{B}}^{-1} \text{CVP}(\mathbf{y}, \mathcal{L}(\mathbf{B}))_i \bmod p/\pi_i)$, for $i = 1, \dots, \ell$.

2.3 Cryptographic Definitions

Definition 10 (M-LWE Problem [5]). The M-LWE samples $(\mathbf{a}_i, b_i = \mathbf{a}_i^T \mathbf{s} + e_i)$ are drawn from the M-LWE distribution $A_{\mathbf{s}, \beta}$ over $R_q^k \times R_q$. Here, $\mathbf{a}_i \leftarrow R_q^k$ is chosen uniformly, $\mathbf{s} \leftarrow \beta_\eta^k$ is common to all samples, and $e_i \leftarrow \beta_\eta$ is independent for each sample. Given m M-LWE samples, the decision-M-LWE problem involves distinguishing $A_{\mathbf{s}, \beta}$ from the uniform distribution on $R_q^k \times R_q$,

while the search- M -LWE problem seeks to recover the secret \mathbf{s} . For an algorithm A , we define the advantage of an adversary as $\text{Adv}_{m,k,\eta}^{\text{M-LWE}}(A) =$

$$\left| \Pr \left(b' = 1 : \begin{array}{l} \mathbf{A} \leftarrow R_q^{m \times k}; (\mathbf{s}, \mathbf{e}) \leftarrow \beta_\eta^k \times \beta_\eta^m \\ \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}; b' \leftarrow A(\mathbf{A}, \mathbf{b}) \end{array} \right) - \Pr \left(b' = 1 : \begin{array}{l} \mathbf{A} \leftarrow R_q^{m \times k}; \\ \mathbf{b} \leftarrow R_q^m; b' \leftarrow A(\mathbf{A}, \mathbf{b}) \end{array} \right) \right|$$

Definition 11 (Public-Key Encryption (PKE) [5]). A public-key encryption scheme $\text{PKE} = (\text{KeyGen}, \text{Enc}, \text{Dec})$ consists of a triple of probabilistic algorithms along with a message space \mathcal{M} . The key-generation algorithm KeyGen produces a pair (pk, sk) , which includes a public key and a secret key. The encryption algorithm Enc takes the public key pk and a message $m \in \mathcal{M}$ to generate a ciphertext c . Finally, the deterministic decryption algorithm Dec uses the secret key sk and the ciphertext c to output either a message $m \in \mathcal{M}$ or a special symbol \perp to indicate rejection. We say that the scheme is $(1 - \delta)$ -correct if $\mathbb{E}[\max_{m \in \mathcal{M}} \Pr[\text{Dec}(sk, \text{Enc}(pk, m)) = m]] \geq 1 - \delta$, where the expectation is taken over (pk, sk) and the probability is taken over the random coins of Enc .

Definition 12 (IND-CPA and IND-CCA [5]). We revisit the standard security notions for public-key encryption, specifically indistinguishability under chosen-ciphertext attacks (IND-CCA) and chosen-plaintext attacks (IND-CPA). The advantage of an adversary A is defined as

$$\text{Adv}_{\text{PKE}}^{\text{CCA}}(A) = \left| \Pr \left(\begin{array}{l} (pk, sk) \leftarrow \text{KeyGen}(); \\ b = b' : \begin{array}{l} (m_0, m_1, s) \leftarrow A^{\text{DEC}(\cdot)}(pk); \\ b \leftarrow \{0, 1\}; c^* \leftarrow \text{Enc}(pk, m_b); \\ b' \leftarrow A^{\text{DEC}(\cdot)}(s, c^*); \end{array} \end{array} \right) - 1/2 \right| \quad (10)$$

where the decryption oracle is defined as $\text{DEC}(\cdot) = \text{Dec}(sk, \cdot)$. We also require that $|m_0| = |m_1|$ and that in the second phase, the adversary A is not permitted to query $\text{DEC}(\cdot)$ with the challenge ciphertext c^* . The advantage $\text{Adv}_{\text{PKE}}^{\text{CPA}}(A)$ of an adversary A is defined as $\text{Adv}_{\text{PKE}}^{\text{CCA}}(A)$, provided that A cannot query $\text{DEC}(\cdot)$.

Definition 13 (DFR and CER). Given a message $m \in \mathcal{M}$, the Decryption Failure Rate (DFR) is denoted as $\delta \triangleq \Pr(\hat{m} \neq m)$, where \hat{m} is the decryption of c where $c = \text{Enc}(pk, m)$. The communication cost refers to the ciphertext expansion rate (CER):

$$\rho = \frac{\# \text{ of bits in } c}{\# \text{ of bits in } m}, \quad (11)$$

i.e., the ratio of the ciphertext size to the plaintext size.

2.4 Kyber's IND-CPA-Secure Encryption

Each message $m \in \{0, 1\}^n$ can be viewed as a polynomial in R with coefficients in $\{0, 1\}$. We recall $\text{Kyber.CPA} = (\text{KeyGen}; \text{Enc}; \text{Dec})$ [4] as described in Algorithms 2.4.1 to 2.4.3. The values of δ , CER, and $(q, k, \eta_1, \eta_2, d_u, d_v)$ are given in Table 2. Note that the parameters (q, k, η_1, η_2) determine the security level of Kyber, while the parameters (d_u, d_v) describe the ciphertext compression rate.

Algorithm 2.4.1 Kyber.CPA.KeyGen(): key generation

- 1: $\psi, \sigma \leftarrow \{0, 1\}^{256}$
 - 2: $\mathbf{A} \sim R_q^{k \times k} := \text{Sam}(\psi)$
 - 3: $(\mathbf{s}, \mathbf{e}) \sim \beta_{\eta_1}^k \times \beta_{\eta_1}^k := \text{Sam}(\sigma)$
 - 4: $\mathbf{t} := \mathbf{A}\mathbf{s} + \mathbf{e}$
 - 5: **return** $(pk := (\mathbf{t}, \psi), sk := \mathbf{s})$
-

Algorithm 2.4.2 Kyber.CPA.Enc $(pk = (\mathbf{t}, \psi), m \in \{0, 1\}^n)$

- 1: $r \leftarrow \{0, 1\}^{256}$
 - 2: $\mathbf{A} \sim R_q^{k \times k} := \text{Sam}(\psi)$
 - 3: $(\mathbf{r}, \mathbf{e}_1, e_2) \sim \beta_{\eta_1}^k \times \beta_{\eta_2}^k \times \beta_{\eta_2} := \text{Sam}(r)$
 - 4: $\mathbf{u} := Q_{\text{Kyber}, 2^{d_u}}(\mathbf{A}^T \mathbf{r} + \mathbf{e}_1)$
 - 5: $v := Q_{\text{Kyber}, 2^{d_v}}(\mathbf{t}^T \mathbf{r} + e_2 + \lceil q/2 \rceil \cdot m)$
 - 6: **return** $c := (\text{Index}_{2^{d_u}}(\mathbf{u}), \text{Index}_{2^{d_v}}(v))$
-

Algorithm 2.4.3 Kyber.CPA.Dec $(sk = \mathbf{s}, c = (\mathbf{u}, v))$

- 1: $\mathbf{u} := C_{2^{d_u}}(\text{Index}_{2^{d_u}}(\mathbf{u}))$
 - 2: $v := C_{2^{d_v}}(\text{Index}_{2^{d_v}}(v))$
 - 3: **return** $\text{Compress}_q(v - \mathbf{s}^T \mathbf{u}, 1)$
-

Table 2. Parameters of Kyber [25]

	k	q	η_1	η_2	d_u	d_v	DFR	CER	Plaintext Size
KYBER512	2	3329	3	2	10	4	2^{-138}	24	256 bits
KYBER768	3	3329	2	2	10	4	2^{-164}	34	256 bits
KYBER1024	4	3329	2	2	11	5	2^{-174}	49	256 bits

2.5 Kyber with Optimal Quantization

The choice of quantization affects the distribution of (c_v, c_u) and thus the decoding noise. [20] shows that Kyber’s quantizer Q_{Kyber} is suboptimal, as it does not minimize the mean squared values of (c_v, c_u) . Replacing it with the MMSE-optimal Lloyd-Max quantizer Q_{MMSE} improves the DFR without affecting Kyber’s security, which is independent of the quantization method and noise level.

Table 3 presents DFR bounds under different quantizers. Using code from [14] and the Lloyd-Max noise distribution, we compute the DFR numerically. A noticeable gap emerges between our results and the CLT-based asymptotic bound in [20], indicating that CLT-based analyses may underestimate DFR. Still, both bounds confirm that improved quantization reduces DFR.

2.6 Coded Kyber and CLT Assumption

Kyber decryption decoding problem can be expressed as [4]

$$y = v - \mathbf{s}^T \mathbf{u} = \lceil q/2 \rceil \cdot m + n_e, \quad (12)$$

Table 3. DFR bounds: Kyber Compression Q_{Kyber} vs. Lloyd-Max Q_{MMSE}

Source	[25]	[20]	This work
Quantization	Q_{Kyber}	Q_{MMSE}	Q_{MMSE}
Bound type	Numerical	CLT	Numerical
KYBER512	2^{-138}	2^{-150}	2^{-142}
KYBER768	2^{-164}	2^{-177}	2^{-169}
KYBER1024	2^{-174}	2^{-196}	2^{-190}

where n_e is the decryption decoding noise

$$n_e = v - \mathbf{s}^T \mathbf{u} - \lceil q/2 \rceil \cdot m = \mathbf{e}^T \mathbf{r} + e_2 + c_v - \mathbf{s}^T (\mathbf{e}_1 + \mathbf{c}_u), \quad (13)$$

where (c_v, \mathbf{c}_u) refers to the quantization noises produced by the quantization Q_{Kyber} in Algorithm 2.4.2. Due to n_e , Kyber decryption has a failure rate.

From the information theory perspective, (12) can be viewed as an *uncoded* 2-PAM [28], which has been generalized to the coded cases [19][20][23]:

$$y = \lceil q/p \rceil \cdot \text{ENC}(m) + n_e, \quad (14)$$

where $p \in \mathbb{Z}$, $m \leftarrow \{0, 1\}^K$, and $\text{ENC}(m) : \{0, 1\}^K \rightarrow \mathbb{Z}_p^n$ represents an encoder. For example, [19] uses a lattice encoder, [20] uses a binary BCH encoder, and [23] uses a Q-ary BCH encoder. The advantage of coded Kyber is the reduced CER, since more information bits can be encrypted to a single ciphertext.

Independence/CLT Assumption in [19][20][23]: To estimate the DFR of coded Kyber, existing schemes assume that the entries in n_e are mutually independent. This assumption relies on the CLT, i.e., for a certain variance σ_e^2 ,

$$\mathbf{e}^T \mathbf{r} - \mathbf{s}^T (\mathbf{e}_1 + \mathbf{c}_u) \rightarrow \mathcal{N}(0, \sigma_e^2 \mathbf{I}_n), \text{ as } k \cdot n \rightarrow \infty \quad (15)$$

An open question is whether we can develop a coded Kyber scheme without relying on the CLT assumption on n_e . We will address this question in the remainder of the paper.

3 Uncoded P_ℓ -Kyber: Kyber with Packed Ciphertexts

In this section, we first present an ℓ -layer Kyber following PVW approach [26]. We then turn this into an IND-CCA KEM and analyze its key and ciphertext sizes as well as its DFR and CER compared to original Kyber.

3.1 IND-CPA-Secure Encryption

We consider the idea of *packed ciphertexts* in [26], where a ciphertext c encrypts a vector of ℓ plaintext ring elements $\mathbf{m} = [m_1, \dots, m_\ell]^T \in R_2^\ell$, not just a single ring element $m \in R_2$. In details, the same matrix \mathbf{A} and encryption randomness \mathbf{r} in Algorithm 2.4.2 can be securely reused to encrypt \mathbf{m} , by having ℓ secret-key vectors $\mathbf{S} = [\mathbf{s}_1, \dots, \mathbf{s}_\ell]$. The key generation, encryption and decryption functions of Kyber with ℓ -packed ciphertexts (P_ℓ -Kyber PKE) is given below.

Algorithm 3.1.1 Kyber.Packed.CPA.KeyGen(): key generation

1: $\psi, \sigma \leftarrow \{0, 1\}^{256}$
2: $\mathbf{A} \sim R_q^{k \times k} := \text{Sam}(\psi)$
3: $(\mathbf{S}, \mathbf{E}) \sim \beta_{\eta_1}^{k \times \ell} \times \beta_{\eta_1}^{k \times \ell} := \text{Sam}(\sigma)$
4: $\mathbf{T} := \mathbf{AS} + \mathbf{E}$
5: **return** $(pk := (\mathbf{T}, \psi), sk := \mathbf{S})$

Algorithm 3.1.2 Kyber.Packed.CPA.Enc $(pk = (\mathbf{T}, \psi), \mathbf{m} \in \{0, 1\}^{\ell \times n})$

1: $r \leftarrow \{0, 1\}^{256}$
2: $\mathbf{A} \sim R_q^{k \times k} := \text{Sam}(\psi)$
3: $(\mathbf{r}, \mathbf{e}_1, \mathbf{e}_2) \sim \beta_{\eta_1}^k \times \beta_{\eta_2}^k \times \beta_{\eta_2}^\ell := \text{Sam}(r)$
4: $\mathbf{u} := Q_{\text{MMSE}, 2^{d_u}}(\mathbf{A}^T \mathbf{r} + \mathbf{e}_1)$
5: $\mathbf{v} := Q_{\text{MMSE}, 2^{d_v}}(\mathbf{T}^T \mathbf{r} + \mathbf{e}_2 + \lceil q/2 \rceil \cdot \mathbf{m})$
6: **return** $c := (\text{Index}_{2^{d_u}}(\mathbf{u}), \text{Index}_{2^{d_v}}(\mathbf{v}))$

Algorithm 3.1.3 Kyber.Packed.CPA.Dec $(sk = \mathbf{S}, c = (\mathbf{u}, \mathbf{v}))$

1: $\mathbf{u} := C_{2^{d_u}}(\text{Index}_{2^{d_u}}(\mathbf{u}))$
2: $\mathbf{v} := C_{2^{d_v}}(\text{Index}_{2^{d_v}}(\mathbf{v}))$
3: **return** $\text{Compress}_q(\mathbf{v} - \mathbf{S}^T \mathbf{u}, 1)$

Correctness. Let δ_ℓ be the DFR of P_ℓ -Kyber PKE. We show below the correctness of the encryption scheme described in Algorithms 3.1.1 to 3.1.3.

Lemma 1 (Correctness of P_ℓ -Kyber PKE). *The DFR is bounded by*

$$\delta_\ell \leq \ell \cdot \delta, \quad (16)$$

where δ is the DFR of the unpacked Kyber in Table 3.

Proof. Let $\mathbf{n}_e = [n_0, \dots, n_{\ell-1}]^T \in R_q^\ell$ be the decoding noise of P_ℓ -Kyber. Similar to (13), we can write \mathbf{n}_e as

$$\mathbf{n}_e = \mathbf{v} - \mathbf{S}^T \mathbf{u} - \lceil q/2 \rceil \cdot \mathbf{m} = \mathbf{E}^T \mathbf{r} + \mathbf{e}_2 + \mathbf{c}_v - \mathbf{S}^T (\mathbf{e}_1 + \mathbf{c}_u), \quad (17)$$

where $(\mathbf{c}_v, \mathbf{c}_u)$ are the quantization noises. Using the union bound, we obtain

$$\delta_\ell = \Pr(\|\mathbf{n}_e\|_\infty \geq \lceil q/4 \rceil) \leq \sum_{i=0}^{\ell-1} \Pr(\|n_i\|_\infty \geq \lceil q/4 \rceil) = \ell \cdot \delta. \quad (18)$$

Remark 1. The DFR of P_ℓ -Kyber increases with ℓ but can be reduced using Q_{MMSE} . Its key benefit is the low CER, denoted as ρ_ℓ :

$$\rho_\ell = \frac{knd_u + \ell nd_v}{N} = \frac{kdu}{\ell} + d_v. \quad (19)$$

where $N = n \cdot \ell$ is the plaintext size (in bits). Table 1 shows $(\delta = \delta_\ell, \rho = \rho_\ell)$ as a function of ℓ (refer to the column labeled “This Work – Uncoded”). P₈-KYBER1024 reduces CER by 79% and DFR by 2^{13} , relative to KYBER1024.

Security. We will prove that the encryption scheme defined above is IND-CPA secure under the M-LWE hardness assumption.

Lemma 2 (IND-CPA Security of P_ℓ -Kyber PKE). *For any adversary A , there exists an adversary B such that $\text{Adv}_{P_\ell\text{-Kyber}}^{\text{CPA}}(A) \leq (\ell + 1) \cdot \text{Adv}_{k+\ell, k, \eta}^{\text{M-LWE}}(B)$.*

Proof. Let A be an adversary that is executed in the IND-CPA security experiment which we call game G_0 , i.e., $\text{Adv}_{P_\ell\text{-Kyber}}^{\text{CPA}} = |\Pr(b = b' \text{ in game } G_0) - 1/2|$.

In game G_1 , the ℓ column vectors in the public key \mathbf{T} are simultaneously substituted with ℓ uniform random vectors. It is possible to verify that there exists an adversary B with the same running time as that of A such that

$$|\Pr(b = b' \text{ in game } G_0) - \Pr(b = b' \text{ in game } G_1)| \leq \ell \text{Adv}_{k, k, \eta}^{\text{M-LWE}}(B) \leq \ell \text{Adv}_{k+\ell, k, \eta}^{\text{M-LWE}}(B), \quad (20)$$

where the second inequality holds since the adversary B will have access to more samples, in particular from k to $k + \ell$.

In game G_2 , the vectors \mathbf{u} and \mathbf{v} used in the generation of the challenge ciphertext are simultaneously substituted with uniform random vectors. Again, there exists an adversary B with the same running time as that of A with

$$|\Pr(b = b' \text{ in game } G_1) - \Pr(b = b' \text{ in game } G_2)| \leq \text{Adv}_{k+\ell, k, \eta}^{\text{M-LWE}}(B). \quad (21)$$

Note that in game G_2 , the value \mathbf{v} from the challenge ciphertext is independent of bit b and therefore $\Pr(b = b' \text{ in game } G_2) = 1/2$. Collecting the probabilities in (20) and (21) yields the required bound.

3.2 The CCA-Secure KEM

Let $G : \{0, 1\}^* \rightarrow \{0, 1\}^{(\ell+1) \times 256}$ and $H : \{0, 1\}^* \rightarrow \{0, 1\}^{\ell \times 256}$ be hash functions. Given $z \leftarrow \{0, 1\}^{\ell \times 256}$, along the same line as [5], a KEM is obtained by applying a KEM variant of the Fujisaki–Okamoto (FO) transform [16] to the P_ℓ -Kyber encryption scheme. We make explicit the randomness r in the Enc algorithm.

Algorithm 3.2.1 `Kyber.Packed.Encaps(pk = (T, ψ))`

- 1: $\mathbf{m} \leftarrow \{0, 1\}^{256 \times \ell}$
 - 2: $(\hat{K}, r) := G(H(pk), \mathbf{m})$
 - 3: $(\mathbf{u}, \mathbf{v}) := \text{Kyber.Packed.CPA.Enc}(pk = (\mathbf{T}, \psi), \mathbf{m}; r)$
 - 4: $c := (\mathbf{u}, \mathbf{v})$
 - 5: $K := H(\hat{K}, H(c))^3$
 - 6: **return** (c, K)
-

³ $H(c)$ was used in [4][5] to simplify the implementation with non-incremental hash APIs. We can use c in place of $H(c)$, as referenced in [16][25]. Another difference between [5] and [4] [25] is that a third hash function, Key Derivation Function (KDF), is used to compute K , i.e., $K := \text{KDF}(\hat{K}, H(c))$ in [4]. Since these small tweaks don't affect the security and DFR analysis of Kyber, we follow the original design in [5].

Algorithm 3.2.2 Kyber.Packed.Decaps($sk = (\mathbf{S}, z, \mathbf{T}, \psi), c = (\mathbf{u}, \mathbf{v})$)

```
1:  $\mathbf{m}' := \text{Kyber.Packed.CPA.Dec}(\mathbf{S}, (\mathbf{u}, \mathbf{v}))$ 
2:  $(\hat{K}', r') := G(H(pk), \mathbf{m}')$ 
3:  $(\mathbf{u}', \mathbf{v}') := \text{Kyber.Packed.CPA.Enc}(pk = (\mathbf{T}, \psi), \mathbf{m}'; r')$ 
4: if  $(\mathbf{u}', \mathbf{v}') = (\mathbf{u}, \mathbf{v})$  then
5:   return  $K := H(\hat{K}', H(c))$ 
6: else
7:   return  $K := H(z, H(c))$ 
8: end if
```

Correctness. If Kyber.Packed.CPA is $(1 - \delta_\ell)$ -correct and G is a random oracle, then P_ℓ -Kyber is $(1 - \delta_\ell)$ -correct [16].

Security. We provide the concrete security bounds from [5][16] which proves P_ℓ -Kyber KEM's CCA-security, when G and H are modelled as random oracles.

Lemma 3 (IND-CCA Secure KEM [16, Theo. 3.2 and 3.4]). *For any classical adversary A that makes at most q_{RO} queries to the random oracles H and G , as well as q_D queries to the decryption oracle, there exists an adversary B such that*

$$\text{Adv}_{P_\ell\text{-Kyber}}^{\text{CCA}}(A) \leq 3\text{Adv}_{P_\ell\text{-Kyber}}^{\text{CPA}}(B) + q_{RO} \cdot \delta_\ell + \frac{3q_{RO}}{2^{256 \times \ell}}. \quad (22)$$

Lemma 4 (IND-CCA Secure KEM [4, Theo. 3][5, Theo. 4]). *For any quantum adversary A that makes at most q_{RO} queries to the quantum random oracles H and G , as well as at most q_D (classical) queries to the decryption oracle, there exists a quantum adversary B such that*

$$\text{Adv}_{P_\ell\text{-Kyber}}^{\text{CCA}}(A) \leq 8q_{RO}^2 \cdot \delta_\ell + 4q_{RO} \sqrt{(\ell + 1) \cdot \text{Adv}_{k+\ell, k, \eta}^{\text{M-LWE}}(B)}. \quad (23)$$

3.3 Parameter Sets

P_ℓ -Kyber KEM adopts the same parameters $(q, k, \eta_1, \eta_2, d_u, d_v)$ as Kyber KEM [25], shown in Table 2. Key and ciphertext sizes are summarized in Table 4, alongside Kyber KEM parameters for encapsulating ℓ messages. P_ℓ -Kyber achieves smaller sizes, especially for large ℓ . Its computational cost is also lower, as \mathbf{u} (or \mathbf{u}') is computed only once in Algorithms 3.2.1 and 3.2.2.

Table 4. Sizes (in bytes) of keys and ciphertexts: P_ℓ -Kyber KEM vs. Kyber KEM

	m	pk	sk	c
P_ℓ -Kyber KEM	$n\ell/8$	$12kn\ell/8 + 32^1$	$24kn\ell/8 + 32\ell + 32^1$	$d_u kn/8 + d_v n\ell/8$
Kyber KEM	$n\ell/8$	$12kn\ell/8 + 32\ell$	$24kn\ell/8 + 64\ell$	$d_u kn\ell/8 + d_v n\ell/8$

¹ In P_ℓ -Kyber, the random seed ψ only need to be transmitted once.

In summary, the P_ℓ -Kyber KEM is a natural extension of the original Kyber KEM [4][25], supporting the encapsulation of $\ell \geq 1$ secrets and a MMSE

quantization Q_{MMSE} . With $\ell = 1$ and a non-MMSE quantization Q_{Kyber} , the P_ℓ -Kyber KEM reduces to the original Kyber KEM. The advantages of P_ℓ -Kyber KEM are twofold: its CER approaching a constant value, d_v , as ℓ increases, while its DFR can be evaluated numerically. Note that the DFR analysis of current CER-oriented approaches [23][19][20] relies on the CLT assumption. However, the downside of P_ℓ -Kyber is its DFR increases linearly with the value of ℓ . Since a high DFR will impact the security bounds in Lemmas 3 and 4, the value of ℓ is bounded, e.g., $\ell \leq 16$ in P_ℓ -KYBER512. In the next section, we will show how to reduce the DFR of the P_ℓ -Kyber KEM by employing lattice codes, which enables the packing of significantly more plaintexts than the uncoded version.

4 Lattice-Coded P_ℓ -Kyber

We now propose our lattice packing approaches to further reduce the DFR of multi-layer Kyber introduced in the previous section. The additional complexities of lattice encoding techniques are provided at the end of this section too.

4.1 Lattice Vertical Encoding and Lattice Packing

The decoding model of uncoded P_ℓ -Kyber can be expressed as

$$\mathbf{Y} = \lceil q/2 \rceil \cdot \mathbf{m} + \mathbf{n}_e, \text{ where} \quad (24)$$

$$\mathbf{n}_e = \mathbf{E}^T \mathbf{r} + \mathbf{e}_2 + \mathbf{c}_v - \mathbf{S}^T (\mathbf{e}_1 + \mathbf{c}_u). \quad (25)$$

Let $\mathbf{n}_e = [n_0, \dots, n_{\ell-1}]^T \in R_q^\ell$ be the decoding noise, where each ring element n_i can be further interpreted as a row vector of integer coefficients $n_{i,j}$, i.e.,

$$n_i = [n_{i,0}, n_{i,1}, \dots, n_{i,n-1}], \quad 0 \leq i \leq \ell - 1.$$

It is more convenient to represent \mathbf{n}_e in matrix form:

$$\phi(\mathbf{n}_e) = \begin{bmatrix} n_{0,0} & n_{0,1} & \cdots & n_{0,n-1} \\ n_{1,0} & n_{1,1} & \cdots & n_{1,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ n_{\ell-1,0} & n_{\ell-1,1} & \cdots & n_{\ell-1,n-1} \end{bmatrix}. \quad (26)$$

where ϕ is given in Definition 4. We can also represent \mathbf{m} in matrix form:

$$\phi(\mathbf{m}) = \begin{bmatrix} m_{0,0} & m_{0,1} & \cdots & m_{0,n-1} \\ m_{1,0} & m_{1,1} & \cdots & m_{1,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ m_{\ell-1,0} & m_{\ell-1,1} & \cdots & m_{\ell-1,n-1} \end{bmatrix}, \quad (27)$$

where $\mathbf{m} = [m_0, \dots, m_{\ell-1}]^T \in R_q^\ell$ is the ℓ messages, and m_i can be further interpreted as a row vector of coefficients $m_{i,j} \in \mathbb{Z}_2$, i.e., $m_i = [m_{i,0}, m_{i,1}, \dots, m_{i,n-1}]$.

By substituting (26) and (27) to (24), the decoding model \mathbf{Y} can be conveniently expressed in matrix form $\phi(\mathbf{Y})$:

$$\left\lfloor \frac{q}{2} \right\rfloor \cdot \underbrace{\begin{bmatrix} m_{0,0} & m_{0,1} & \cdots & m_{0,n-1} \\ m_{1,0} & m_{1,1} & \cdots & m_{1,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ m_{\ell-1,0} & m_{\ell-1,1} & \cdots & m_{\ell-1,n-1} \end{bmatrix}}_{\phi(\mathbf{m})} + \underbrace{\begin{bmatrix} n_{0,0} & n_{0,1} & \cdots & n_{0,n-1} \\ n_{1,0} & n_{1,1} & \cdots & n_{1,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ n_{\ell-1,0} & n_{\ell-1,1} & \cdots & n_{\ell-1,n-1} \end{bmatrix}}_{\phi(\mathbf{n}_e)} \begin{matrix} \text{i.i.d. RVs} \\ \\ \\ \text{depend.} \\ \text{RVs} \end{matrix} \quad (28)$$

Lemma 5 (Vertical Decoding Noise). *In (28), the entries within each column of \mathbf{n}_e are independent and identically distributed (i.i.d.) random variables.*

Proof. Recalling that $\mathbf{n}_e = \mathbf{E}^T \mathbf{r} + \mathbf{e}_2 + \mathbf{c}_v - \mathbf{S}^T (\mathbf{e}_1 + \mathbf{c}_u)$. Without loss of generality, we study the distribution of the first column in $\phi(\mathbf{n}_e)$, i.e., $\Pr(n_{0,0}, \dots, n_{\ell-1,0})$. For $0 \leq i \leq \ell - 1$, we observe that $n_{i,0}$ is generated by the same realization of $(\mathbf{r}, \mathbf{e}_1 + \mathbf{c}_u)$, denoted as (\mathbf{a}, \mathbf{b}) . We can interpret $n_{i,0}$ as a deterministic function of random variables with fixed parameters (\mathbf{a}, \mathbf{b}) :

$$n_{i,0} = g_{(\mathbf{a}, \mathbf{b})}(E_i, \mathbf{s}_i, e_{2,i}, c_{v,i}), \quad (29)$$

where E_i and \mathbf{s}_i are the i -th columns in \mathbf{E} and \mathbf{S} , respectively. And $e_{2,i}$ and $c_{v,i}$ represent the i -th elements in \mathbf{e}_2 and \mathbf{c}_v , respectively. Since $(E_i, \mathbf{s}_i, e_{2,i}, c_{v,i})$ are mutually independent for $0 \leq i \leq \ell - 1$, $\{n_{i,0}\}_{i=0}^{\ell-1}$ are i.i.d. random variables.

Remark 2. The current encoding schemes for (R/M-)LWE can be viewed as *Horizontal Encoding* (H-Enc) [23][19][22][15], where the rows of $\phi(\mathbf{m})$ are encoded. The major issue of H-Enc is that the elements in each row of $\phi(\mathbf{n}_e)$ are dependent. The DFR analysis has to assume that the noise coefficients in each row are mutually independent (CLT), which may result in an underestimated DFR. Given that each column of $\phi(\mathbf{n}_e)$ consists of i.i.d. RVs, it is natural to encode $\phi(\mathbf{m})$ column-wise, thereby circumventing reliance on the CLT assumption.

Definition 14 (Lattice-Based Vertical Encoding (LV-Enc)). *Given $\mathbf{m} \leftarrow \mathcal{M}_{p,\ell}^n$, the lattice encoder and decoder are defined by*

$$\begin{aligned}
\text{Enc}_\Lambda &: \mathcal{M}_{p,\ell}^n \rightarrow (\mathcal{L}(\hat{\mathbf{B}}) \cap \mathbb{Z}_p^\ell)^n \\
\text{Dec}_\Lambda &: (\mathcal{L}(\hat{\mathbf{B}}) \cap \mathbb{Z}_p^\ell)^n \rightarrow \mathcal{M}_{p,\ell}^n
\end{aligned} \quad (30)$$

where $\text{Enc}_\Lambda(\phi(\mathbf{m})) := \phi(\hat{\mathbf{m}}) = \hat{\mathbf{B}}\phi(\mathbf{m}) \bmod p$ encodes the n columns of $\phi(\mathbf{m})$ into n lattice points $\phi(\hat{\mathbf{m}})$ in a column-wise manner, and $\text{Dec}_\Lambda(\phi(\hat{\mathbf{m}})) := \phi(\mathbf{m}) = \text{CVP}_{\text{HS}}(\phi(\hat{\mathbf{m}}))$ takes the input of $\phi(\hat{\mathbf{m}})$ and returns $\phi(\mathbf{m})$. The notations of the lattice $\mathcal{L}(\hat{\mathbf{B}})$, the matrix $\hat{\mathbf{B}} \in \mathbb{Z}^{\ell \times \ell}$, the message space $\mathcal{M}_{p,\ell}$, the decoder $\text{CVP}_{\text{HS}}(\cdot)$, and the hypercube shaping $\mathcal{L}(\hat{\mathbf{B}}) \cap \mathbb{Z}_p^\ell$ are given in Section 2.2.

To gain more insight, the coded version of (28) is described by $\phi(\mathbf{Y}) =$

$$\begin{aligned}
& \left[\frac{q}{p} \right] \cdot \hat{\mathbf{B}} \cdot \underbrace{\begin{bmatrix} m_{0,0} & \underbrace{m_{0,1} \cdots m_{0,n-1}}_{\phi(\hat{\mathbf{m}})} \\ m_{1,0} & \underbrace{m_{1,1} \cdots m_{1,n-1}} \\ \vdots & \vdots \\ m_{\ell-1,0} & \underbrace{m_{\ell-1,1} \cdots m_{\ell-1,n-1}} \end{bmatrix}}_{\phi(\hat{\mathbf{m}})} \pmod{p} + \underbrace{\begin{bmatrix} n_{0,0} & \underbrace{n_{0,1} \cdots n_{0,n-1}}_{\phi(\mathbf{n}_e)} \\ n_{1,0} & \underbrace{n_{1,1} \cdots n_{1,n-1}} \\ \vdots & \vdots \\ n_{\ell-1,0} & \underbrace{n_{\ell-1,1} \cdots n_{\ell-1,n-1}} \end{bmatrix}}_{\phi(\mathbf{n}_e)} \pmod{p} \\
& \hspace{15em} \text{i.i.d. RVs}
\end{aligned} \tag{31}$$

Lattice packing. The distribution of the noise vectors in LV-Enc is bounded by a hypersphere with high probability (We will show this in Lemma 6). Since the addition in (28) is over the modulo q domain, the LV-Enc problem in P_ℓ -Kyber can be viewed as a *lattice packing* problem: an arrangement of non-overlapping spheres within a hypercube \mathbb{Z}_q^ℓ . The model in (28) uses the integer lattice codes $\lfloor q/2 \rfloor \mathbb{Z}_2^\ell$ for packing purposes, which is far from optimal. Even for very small dimensions ℓ , there exists much denser lattice packings than cubic ones.

Definition 15 (Coded P_ℓ -Kyber PKE). *The encryption and decryption of the uncode P_ℓ -Kyber PKE can be easily adapted for the coded version by implementing the following modifications.*

- Coded version of Algorithm 3.1.2
 - input message space: replace $\{0, 1\}^{\ell \times n}$ by $\mathcal{M}_{p,\ell}^n$
 - Step 5: replace \mathbf{m} by $\phi^{-1}(\text{Enc}_\Lambda(\phi(\mathbf{m})))$
- Coded version of Algorithm 3.1.3
 - Step 3: replace $\text{Compress}_q(\mathbf{v} - \mathbf{S}^T \mathbf{u}, 1)$ by $\text{Dec}_\Lambda(\phi(\mathbf{v} - \mathbf{S}^T \mathbf{u}))$

For the choice of $\mathcal{L}(\hat{\mathbf{B}})$, in this work, we consider E8 lattice with $\ell = 8$, Barnes–Wall lattice with $\ell = 16$ (BW16), and Leech lattice with $\ell = 24$ (Leech24)[12]. These lattices provides the best known sphere packing in their dimension ℓ . Since the coefficients in Kyber are integers, we will scale the original generator matrix to an integer matrix and utilize the corresponding $\hat{\mathbf{B}}$.

Correctness. Let $\lambda(p)$ be the length of a shortest non-zero vector in the lattice $\mathcal{L}(\lfloor q/p \rfloor \hat{\mathbf{B}})$. The correct decoding radius of HS-CVP decoder (i.e., packing radius of $\mathcal{L}(\lfloor q/p \rfloor \hat{\mathbf{B}})$) is $\lambda(p)/2$. We show below the correctness of coded P_ℓ -Kyber PKE.

Lemma 6 (DFR of Coded P_ℓ -Kyber PKE).

$$\delta_\ell \leq n \exp\left(-\theta \lambda(p)^2/4 + \ell \log(M_{n_{0,0}^2}(\theta))\right), \tag{32}$$

where $M_X(\theta)$ is the moment generating function of X , defined in Section 2.1.

Proof. We first study the DFR for the first lattice point (the first column in $\hat{\mathbf{m}}$).

$$\delta^{(1)} = \Pr\left(\sum_{i=0}^{\ell-1} n_{i,0}^2 \geq \lambda(p)^2/4\right) \tag{33}$$

Since $n_{i,0}$, for $0 \leq i \leq \ell - 1$ are i.i.d., we have

$$M_{\sum_{i=0}^{\ell-1} n_{i,0}^2}(\theta) = M_{n_{0,0}^2}(\theta)^\ell \quad (34)$$

Using Chernoff bound and union bound, we obtain (32).

Remark 3. We numerically search the optimal θ which satisfies

$$\theta = \arg \min_{\theta' \in \mathbb{R}} \exp\left(-\theta' \lambda(p)^2 / 4 + \ell \log(M_{n_{0,0}^2}(\theta'))\right). \quad (35)$$

The distribution of $n_{0,0}^2$ can be obtained from the Python code in [14]. For demonstration purposes, we plot the distribution of $n_{0,0}^2$ for P_ℓ -KYBER1024 in Fig. 1. For different choices of $\mathcal{L}(\hat{\mathbf{B}})$, the values of $\lambda(p)$ are listed in Table 5.

Plaintext size and CER. Let $K_{p,\ell} = \log_2(|\mathcal{M}_{p,\ell}|)$ denote the information bit length per lattice codeword in (31). According to Definition 8, the plaintext size of coded P_ℓ -Kyber, N (in bits), can be computed by

$$N = n \cdot K_{p,\ell} = n \cdot \sum_{i=1}^{\ell} \log_2(p/\pi_i), \quad (36)$$

where π_i is given in (8), for $i = 1, \dots, \ell$. The CER of coded P_ℓ -Kyber is

$$\rho_\ell = \frac{knd_u + \ell nd_v}{N} = \frac{kdu + \ell dv}{K_{p,\ell}}. \quad (37)$$

Table 5 lists the (δ_ℓ, ρ_ℓ) values for various lattice encoders. In comparison to the values of ρ_ℓ for uncoded P_ℓ -Kyber, we notice that the coded version has a smaller ρ_ℓ . This can be explained by $K_{p,\ell} \geq \ell$, i.e., the uncoded P_ℓ -Kyber embeds ℓ secret bits in each column of $\phi(\mathbf{m})$ in (28), while the coded version encodes $K_{p,\ell}$ secret bits in each column of $\phi(\hat{\mathbf{m}})$ in (31). Coded P_{24} -KYBER1024 reduces CER by 90% and DFR by 2^{107} , relative to KYBER1024.

Security. The security proofs of coded P_ℓ -Kyber PKE and KEM are the same as the uncoded versions and thus omitted.

4.2 Side-Channel Attack, Constant-Time Decoder, and Complexity

The implementation of the lattice decoder may be susceptible to side-channel attacks. In [13], The authors note that the decoding process typically recovers valid codewords more quickly than those containing errors. This timing information can be exploited to differentiate between valid ciphertexts and failing ciphertexts. However, this attack can be mitigated by employing a constant-time decoder. The fundamental idea is to partition the lattice Λ into the cosets of a specific sublattice Λ' . The constant-time decoding problem for Λ can be reduced to the constant-time decoding problem for Λ' . by exhaustively searching through all cosets of Λ' .

In Table 6, we recall the time complexity of existing constant-time lattice decoders in [19]. We count the total numbers of additional-equivalent operations as in [10]. Let $\mathcal{L}(\mathbf{D}_\ell)$ be the ℓ -dimensional checkerboard lattice [12], and $\mathcal{L}(\mathbf{Q}_{24})$ be the Leech quarter lattice [34]. From an engineering perspective, the

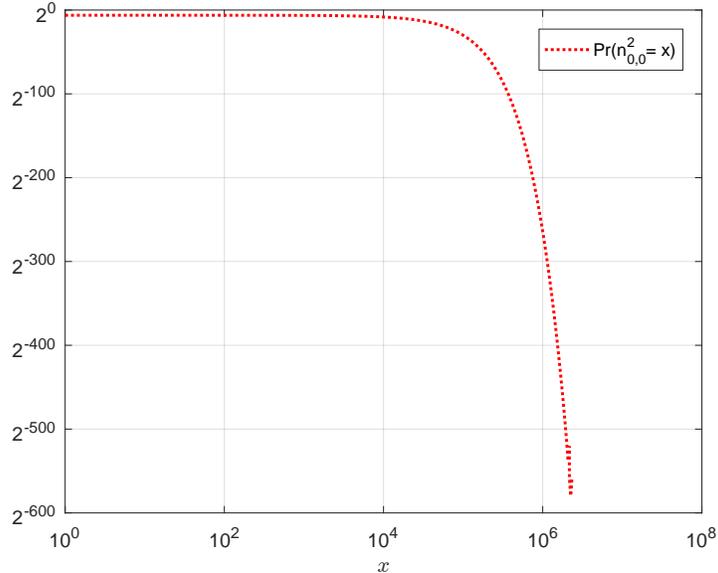


Fig. 1. P_ℓ -KYBER1024: distribution of $n_{0,0}^2$ from the Python code in [14]

energy consumption associated with communication significantly exceeds that of computation. We believe that lattice codes can help reduce the overall energy consumption.

In summary, we demonstrate that lattice-based vertical encoding via ciphertext packing effectively reduces both DFR and CER of the original Kyber, by increasing the plaintext size N , without relying on the CLT assumption for decoding noise.

5 Truncated Lattice-Coded P_ℓ -Kyber: $N = 256$ bits

In practical scenarios, the plaintext size is commonly fixed ($\ell = 1$ or $\ell = 2$), e.g., $N = 256$ bits as opposed to having large $\ell > 2$ in Section 4. In this section, we will show that plaintext and ciphertext size of coded P_ℓ -Kyber can be naturally adapted by truncation.

5.1 Coded P_ℓ -Kyber with Truncated Ciphertext

Let us start by providing the definition of truncation.

Definition 16 (Truncation Function). Let $\mathbf{A} \in \mathbb{F}^{\ell \times n}$ be a matrix over a field \mathbb{F} , where $A = [a_{i,j}]$ and each entry $a_{i,j} \in \mathbb{F}$ for $0 \leq i \leq \ell - 1$, $0 \leq j \leq n - 1$. Let $t \in \mathbb{N}$ be such that $1 \leq t \leq n$. The truncation function Trunc_t is defined as:

$$\text{Trunc}_t(\mathbf{A}) := [a_{i,j}]_{0 \leq i \leq \ell - 1, 0 \leq j \leq t - 1}.$$

That is, $\text{Trunc}_t(\mathbf{A})$ returns a matrix consisting of the first t columns of \mathbf{A} , by removing its last $n - t$ columns.

Table 5. Lattice Codes for P_ℓ -Kyber: using Kyber's $(k, q, \eta_1, \eta_2, d_u, d_v)$

Lattice	Uncoded	Coded		
	\mathbb{Z}^8	E8	BW16	Leech24
ℓ	8	8	16	24
p	2	4	4	8
$\lambda(p)/(2 \lfloor q/2 \rfloor)$	0.5	0.7067	0.7067	0.7067
$K_{p,\ell}$	8	8	20	36
N (in bits)	$8n$ (8 AES keys)	$8n$ (8 AES keys)	$20n$ (20 AES keys)	$36n$ (36 AES keys)
P_ℓ -KYBER512	$\delta_\ell = 2^{-139}$ $\rho_\ell = 6.5$	$\delta_\ell = 2^{-225}$ $\theta = 1.4 \times 10^{-4}$ $\rho_\ell = 6.5$	$\delta_\ell = 2^{-184}$ $\theta = 1.3 \times 10^{-4}$ $\rho_\ell = 4.2$	$\delta_\ell = 2^{-155}$ $\theta = 1.2 \times 10^{-4}$ $\rho_\ell = 3.2$
P_ℓ -KYBER768	$\delta_\ell = 2^{-166}$ $\rho_\ell = 7.8$	$\delta_\ell = 2^{-267}$ $\theta = 1.7 \times 10^{-4}$ $\rho_\ell = 7.8$	$\delta_\ell = 2^{-217}$ $\theta = 1.53 \times 10^{-4}$ $\rho_\ell = 4.7$	$\delta_\ell = 2^{-183}$ $\theta = 1.4 \times 10^{-4}$ $\rho_\ell = 3.5$
P_ℓ -KYBER1024	$\delta_\ell = 2^{-187}$ $\rho_\ell = 10.5$	$\delta_\ell = 2^{-336}$ $\theta = 1.88 \times 10^{-4}$ $\rho_\ell = 10.5$	$\delta_\ell = 2^{-306}$ $\theta = 1.85 \times 10^{-4}$ $\rho_\ell = 6.2$	$\delta_\ell = 2^{-281}$ $\theta = 1.79 \times 10^{-4}$ $\rho_\ell = 4.6$

Table 6. Constant-time lattice decoders: time complexity

Lattice decoder	\mathbb{Z} [4]	E8 [9] [12]	BW16 [12][22]	Leech24 [11][10]	Leech24 [27]
Lattice dimension	1	8	16	24	24
Λ	\mathbb{Z}	$\mathcal{L}(\mathbf{D}_8)$	$\mathcal{L}(\mathbf{D}_{16})$	$\mathcal{L}(\mathbf{D}_{24})$	$\mathcal{L}(\mathbf{Q}_{24})$
# of operations	1	64	2048	786432	≈ 3974

The plaintext and ciphertext size of coded/uncoded P_ℓ -Kyber can be easily adjusted according to a given plaintext size, e.g., $N = 256$ bits. The basic idea is to truncate the ciphertext $\mathbf{v} = Q_{\text{MMSE}, 2d_v}(\mathbf{T}^T \mathbf{r} + \mathbf{e}_2 + \lfloor q/2 \rfloor \cdot \text{Enc}_\Lambda(\mathbf{m}))$. Let $\mathbf{v} = [v_0, \dots, v_{\ell-1}]^T \in R_q^\ell$, where each ring element v_i can be further interpreted as a row vector of integer coefficients $v_{i,j}$, i.e.,

$$v_i = [v_{i,0}, v_{i,1}, \dots, v_{i,n-1}], \quad 0 \leq i \leq \ell - 1.$$

The vector \mathbf{v} can be equivalently expressed in matrix form:

$$\phi(\mathbf{v}) = \begin{bmatrix} v_{0,0} & v_{0,1} & \cdots & v_{0,n-1} \\ v_{1,0} & v_{1,1} & \cdots & v_{1,n-1} \\ \vdots & \vdots & \cdots & \vdots \\ v_{\ell-1,0} & v_{\ell-1,1} & \cdots & v_{\ell-1,n-1} \end{bmatrix}. \quad (38)$$

Due to P_ℓ -Kyber's column-wise decoding structure, i.e., $\text{Dec}_\Lambda(\phi(\mathbf{v} - \mathbf{S}^T \mathbf{u}))$, the last $n - t$ columns of $\phi(\mathbf{v})$ can be removed, resulting in a truncated vector $\hat{\mathbf{v}}$, whose matrix representation is given by:

$$\phi(\hat{\mathbf{v}}) = \text{Trunc}_t(\phi(\mathbf{v})) = \begin{bmatrix} v_{0,0} & v_{0,1} & \cdots & v_{0,t-1} \\ v_{1,0} & v_{1,1} & \cdots & v_{1,t-1} \\ \vdots & \cdots & \vdots & \\ v_{\ell-1,0} & v_{\ell-1,1} & \cdots & v_{\ell-1,t-1} \end{bmatrix}. \quad (39)$$

The corresponding decoding model, i.e., $\phi(\mathbf{Y}) = \text{Trunc}_t(\phi(\mathbf{v} - \mathbf{S}^T \mathbf{u}))$, is given by

$$\left[\frac{q}{p} \right] \cdot \hat{\mathbf{B}} \cdot \underbrace{\begin{bmatrix} m_{0,0} & \overbrace{m_{0,1} \cdots m_{0,t-1}}^{\in \mathcal{M}_{p,\ell}} \\ m_{1,0} & \overbrace{m_{1,1} \cdots m_{1,t-1}} \\ \vdots & \vdots \\ m_{\ell-1,0} & \overbrace{m_{\ell-1,1} \cdots m_{\ell-1,t-1}} \end{bmatrix}}_{\phi(\hat{\mathbf{m}})} \bmod p + \underbrace{\begin{bmatrix} n_{0,0} & \overbrace{n_{0,1} \cdots n_{0,t-1}}^{\text{i.i.d. RVs}} \\ n_{1,0} & \overbrace{n_{1,1} \cdots n_{1,t-1}} \\ \vdots & \vdots \\ n_{\ell-1,0} & \overbrace{n_{\ell-1,1} \cdots n_{\ell-1,t-1}} \end{bmatrix}}_{\phi(\mathbf{n}_e)} \quad (40)$$

The plaintext size is reduced to $N = tK_{p,\ell}$, where $K_{p,\ell}$ is given in Table 5. The time complexity of truncation is $O(\ell(n-t))$.

Definition 17 (P_{t,ℓ}-Kyber PKE). *The encryption and decryption of the uncode P_ℓ-Kyber PKE can be easily adapted for the truncated coded version, denoted by P_{t,ℓ}-Kyber, by implementing the following modifications.*

- Truncated coded version of Algorithm 3.1.2
 - input message space: replace $\{0, 1\}^{256}$ by $\mathcal{M}_{p,\ell}^t$
 - Step 5: $\mathbf{v} := Q_{\text{MMSE}, 2^{d_v}}(\phi^{-1}(\text{Trunc}_t(\phi(\mathbf{T}^T \mathbf{r} + \mathbf{e}_2)) + \lceil q/2 \rceil \cdot \text{Enc}_\Lambda(\phi(\mathbf{m}))))$
- Truncated coded version of Algorithm 3.1.3
 - Step 3: replace $\text{Compress}_q(\mathbf{v} - \mathbf{S}^T \mathbf{u}, 1)$ by $\text{Dec}_\Lambda(\phi(\mathbf{v}) - \text{Trunc}_t(\phi(\mathbf{S}^T \mathbf{u})))$

Correctness. Using Lemma 6 with $n = t$, the DFR of P_{t,ℓ}-Kyber is given by

$$\delta_{t,\ell} \leq t \exp\left(-\theta \lambda(p)^2 / 4 + \ell \log(M_{n_{0,0}^2}(\theta))\right), \quad (41)$$

where $M_X(\theta)$ is the moment generating function of X , defined in Section 2.1.

Plaintext and ciphertext size. For a fixed plaintext size N (in bits), e.g., $N = 256$, one can select the number of packed codewords as $t = N/K_{p,\ell}$, where $K_{p,\ell} = \log_2(|\mathcal{M}_{p,\ell}|)$ is the information bit length per lattice codeword in Table 5. The resulting ciphertext size M , corresponding to the pair (\mathbf{u}, \mathbf{v}) , is given by:

$$M = knd_u + tK_{p,\ell}d_v = knd_u + Nd_v. \quad (42)$$

Given $N = 256$ bits and same (k, d_u, d_v) , the ciphertext size of P_{t,ℓ}-Kyber is the same as that of standard Kyber. The CER of P_{t,ℓ}-Kyber is given by

$$\rho_{t,\ell} = \frac{M}{N} = \frac{knd_u + tK_{p,\ell}d_v}{N}. \quad (43)$$

Security. The ciphertext of P_{t,ℓ}-Kyber is derived by deterministically discarding $(n-t)\ell$ coefficients from the coded P_ℓ-Kyber ciphertext, thereby preserving at least the same level of security as coded P_ℓ-Kyber.

5.2 CER Reduction Through Tighter Compression Parameters

Since $\delta_{t,\ell} \leq \delta_\ell$, and the values of δ_ℓ are significantly lower than that of standard Kyber (see Table 5), the CER can be reduced by selecting smaller compression parameters (d_u, d_v) .

In Table 7, we evaluate the performance of $P_{t,\ell}$ -Kyber with parameters ($N = 32$ bytes, $t = 32, \ell = 8$), employing the E8 lattice encoder. The table reports DFR and CER values for various (d_u, d_v) configurations. Relative to the original KYBER1024, coded $P_{t,\ell}$ -KYBER1024 achieves a 10.2% reduction in CER and a DFR reduction by a factor of 2^{30} , using $(d_u = 10, d_v = 4)$. If a DFR of 2^{-128} is deemed sufficient, the CER can be further reduced by 16.3% with $(d_u = 9, d_v = 5)$.

Table 7. Parameters of $P_{t,\ell}$ -Kyber with $(\ell = 8, t = 32)$

	k	q	η_1	η_2	d_u	d_v	DFR	CER	Plaintext Size	Ciphertext Size
KYBER1024 [25]	4	3329	2	2	11	5	2^{-174}	49	32 bytes	1568 bytes
$P_{t,\ell}$ -KYBER1024	4	3329	2	2	10	4	2^{-204}	44	32 bytes	1408 bytes
$P_{t,\ell}$ -KYBER1024	4	3329	2	2	9	6	2^{-138}	42	32 bytes	1344 bytes
$P_{t,\ell}$ -KYBER1024	4	3329	2	2	9	5	2^{-128}	41	32 bytes	1312 bytes

For completeness, Table 1 lists the $(\delta = \delta_{t,\ell}, \rho = \rho_{t,\ell})$ values for $P_{t,\ell}$ -Kyber, as shown in the column titled “This Work-Lattice” and the rows labeled “1 – 2 AES keys”. Specifically, for the case of one AES key, we consider parameters $(t = 32, \ell = 8, d_u = 10, d_v = 4)$, and for two AES keys, $(t = 64, \ell = 8, d_u = 10, d_v = 4)$. In both configurations, the E8 lattice encoder is utilized. Notably, we observe that $P_{t,\ell}$ -Kyber achieves the encryption of two AES keys using a ciphertext size of 1536 bytes, whereas KYBER1024 requires 1568 bytes to encapsulate a single AES key. This highlights the inefficiency of the original Kyber encoding and suggests significant room for optimization.

In summary, for a fixed plaintext size of $N = 256$ bits, the proposed $P_{t,\ell}$ -Kyber scheme achieves lower CER and DFR compared to the original Kyber, at the cost of an increased public key size of $12kn\ell/8 + 32$ bytes, as detailed in Table 4. However, since many cryptographic protocols—including Kyber—allow the public key to be pre-stored and reused across multiple encapsulations, the communication overhead introduced by the larger public key becomes negligible as the number of encapsulations grows.

6 Application to Multi-Recipient KEM

We consider the Multi-Recipient Key Encapsulation Mechanism (mKEM) in [17], which securely sends the same session key \mathbf{m} to a group of L recipients. For definitions, syntaxes, and security models of mKEM and mPKE, please refer to Appendix. The construction of an IND-CPA secure mPKE is in most cases a simple modification of an IND-CPA secure PKE to the multi-recipient setting.

The mPKE scheme based on $P_{t,\ell}$ -Kyber ($P_{t,\ell}$ -mPKE) is detailed in Algorithms 6.0.1–6.0.3. A global public matrix $\mathbf{A} \sim R_q^{k \times k} := \text{Sam}(\psi)$ is sampled and

made available to both the sender and all recipients. Each recipient i executes Algorithm 6.0.1 to generate a key pair $(pk_i = \mathbf{T}_i, sk_i = \mathbf{S}_i)$, and forwards pk_i to the sender. The sender aggregates the public keys $\{pk_i = \mathbf{T}_i\}_{i=0}^{L-1}$ and employs Algorithm 6.0.2 to encrypt a session key \mathbf{m} , producing a ciphertext c , which is then distributed to all recipients. Upon receiving c , each recipient i applies Algorithm 6.0.3 with their secret key $sk_i = \mathbf{S}_i$ to recover the session key \mathbf{m} .

Algorithm 6.0.1 $P_{t,\ell}$ -mPKE.CPA.KeyGen(): key generation at Recipient i

- 1: $\psi, \sigma \leftarrow \{0, 1\}^{256}$
 - 2: $\mathbf{A} \sim R_q^{k \times k} := \text{Sam}(\psi)$
 - 3: $(\mathbf{S}_i, \mathbf{E}) \sim \beta_{\eta_1}^{k \times \ell} \times \beta_{\eta_1}^{k \times \ell} := \text{Sam}(\sigma)$
 - 4: $\mathbf{T}_i := \mathbf{A}\mathbf{S}_i + \mathbf{E}$
 - 5: **return** $(pk_i := (\mathbf{T}_i, \psi), sk_i := \mathbf{S}_i)$
-

Algorithm 6.0.2 $P_{t,\ell}$ -mPKE.CPA.Enc $(pk = (\{\mathbf{T}_i\}_{i=0}^{L-1}, \psi), \mathbf{m} \in \{0, 1\}^{\ell \times t})$

- 1: $r \leftarrow \{0, 1\}^{256}$
 - 2: $\mathbf{A} \sim R_q^{k \times k} := \text{Sam}(\psi)$
 - 3: $(\mathbf{r}, \mathbf{e}_1, \mathbf{e}_2) \sim \beta_{\eta_1}^k \times \beta_{\eta_2}^k \times \beta_{\eta_2}^\ell := \text{Sam}(r)$
 - 4: $\mathbf{u} := Q_{\text{MMSE}, 2^{d_u}}(\mathbf{A}^T \mathbf{r} + \mathbf{e}_1)$
 - 5: **for** $i \leftarrow 0$ to $L - 1$ **do**
 - 6: $\mathbf{v}_i := Q_{\text{MMSE}, 2^{d_v}}(\phi^{-1}(\text{Trunc}_t(\phi(\mathbf{T}_i^T \mathbf{r} + \mathbf{e}_2)) + \lceil q/2 \rceil \cdot \text{Enc}_\Lambda(\phi(\mathbf{m}))))$
 - 7: **end for**
 - 8: **return** $c := (\text{Index}_{2^{d_u}}(\mathbf{u}), \text{Index}_{2^{d_v}}(\mathbf{v}_0), \dots, \text{Index}_{2^{d_v}}(\mathbf{v}_{L-1}))$
-

Algorithm 6.0.3 $P_{t,\ell}$ -mPKE.CPA.Dec $(sk_i = \mathbf{S}_i, c = (\mathbf{u}, \mathbf{v}_i))$

- 1: $\mathbf{u} := \mathcal{C}_{2^{d_u}}(\text{Index}_{2^{d_u}}(\mathbf{u}))$
 - 2: $\mathbf{v}_i := \mathcal{C}_{2^{d_v}}(\text{Index}_{2^{d_v}}(\mathbf{v}_i))$
 - 3: **return** $\text{Dec}_\Lambda(\phi(\mathbf{v}_i) - \text{Trunc}_t(\phi(\mathbf{S}_i^T \mathbf{u})))$
-

Correctness. The DFR of $P_{t,\ell}$ -mPKE is same as the $P_{t,\ell}$ -Kyber in Table 7.

Compact Ratio. We recall the notation of compact ratio (CR) defined in [17]:

$$\mu = \frac{L \cdot \text{size of } \mathbf{u} + L \cdot \text{size of } \mathbf{v}_i}{\text{size of } \mathbf{u} + L \cdot \text{size of } \mathbf{v}_i} \approx 1 + \frac{\text{size of } \mathbf{u}}{\text{size of } \mathbf{v}_i} \quad (44)$$

which measures asymptotically how much more compact mPKE is compared to L instances of the original PKE, for a large L . Table 8 presents the values of (δ, μ, N, M) for the proposed $P_{t,\ell}$ -mPKE scheme and those reported in [17]. It can be observed that $P_{t,\ell}$ -mPKE achieves a higher μ , implying improved communication efficiency compared to the scheme in [17]. For a large L , the ciphertext size of $P_{t,\ell}$ -mPKE is about 80% of the scheme in [17].

Security. We will prove that the encryption scheme defined above is IND-CPA secure under the M-LWE hardness assumption.

Table 8. Parameters of $P_{t,\ell}$ -Kyber mPKE with $(\ell = 8, t = 32)$ and L Recipients

	k	q	η_1	η_2	d_u	d_v	DFR	CR	Ciphertext Size
KYBER1024-mPKE [17]	4	3329	2	2	11	5	2^{-174}	9.8	$1408 + 160L$ bytes
$P_{t,\ell}$ -KYBER1024-mPKE	4	3329	2	2	10	4	2^{-204}	11	$1280 + 128L$ bytes

Definition 18 (IND-CPA and IND-CCA of mPKE [17]). We revisit the security notions for mPKE encryption, specifically indistinguishability under chosen-ciphertext attacks (IND-CCA) and chosen-plaintext attacks (IND-CPA). The advantage of an adversary A is defined as

$$\text{Adv}_{\text{mPKE},L}^{\text{CCA}}(A) = \left| \Pr \left(\begin{array}{l} \{\text{pk}_i, \text{sk}_i\}_{i \in [L]} \leftarrow \text{KeyGen}(); \\ (m_0, m_1, s) \leftarrow \text{A}^{\text{DEC}(\cdot)}(\{\text{pk}_i\}_{i \in [L]}); \\ b \leftarrow \{0, 1\}; c^* \leftarrow \text{Enc}(\{\text{pk}_i\}_{i \in [L]}, m_b); \\ b' \leftarrow \text{A}^{\text{DEC}(\cdot)}(s, c^*); \end{array} \right) - 1/2 \right| \quad (45)$$

where the decryption oracle is defined as $\text{DEC}(\cdot) = \text{Dec}(\text{sk}, \cdot)$. We also require that $|m_0| = |m_1|$ and that in the second phase, the adversary A is not permitted to query $\text{DEC}(\cdot)$ with the challenge ciphertext c^* . The advantage $\text{Adv}_{\text{mPKE},L}^{\text{CPA}}(A)$ of an adversary A is defined as $\text{Adv}_{\text{mPKE},L}^{\text{CCA}}(A)$, provided that A cannot query $\text{DEC}(\cdot)$.

Lemma 7 (IND-CPA Security of $P_{t,\ell}$ -mPKE). For any adversary A , there exists an adversary B such that $\text{Adv}_{P_{t,\ell}\text{-mPKE},L}^{\text{CPA}}(A) \leq L(\ell + 1) \cdot \text{Adv}_{k+\ell,k,\eta}^{\text{M-LWE}}(B)$.

Proof. Let A be an adversary that is executed in the IND-CPA security experiment which we call game G_0 , i.e., $\text{Adv}_{P_{t,\ell}\text{-mPKE},L}^{\text{CPA}} = |\Pr(b = b' \text{ in game } G_0) - 1/2|$.

In game G_1 , the ℓ column vectors in each public key \mathbf{T}_i are simultaneously substituted with ℓ uniform random vectors. It is possible to verify that there exists an adversary B with the same running time as that of A such that

$$|\Pr(b = b' \text{ in game } G_0) - \Pr(b = b' \text{ in game } G_1)| \leq L\ell \cdot \text{Adv}_{k,k,\eta}^{\text{M-LWE}}(B). \quad (46)$$

In game G_2 , the vectors \mathbf{u} and \mathbf{v}_i used in the generation of the challenge ciphertext are simultaneously substituted with uniform random vectors. Again, there exists an adversary B with the same running time as that of A with

$$|\Pr(b = b' \text{ in game } G_1) - \Pr(b = b' \text{ in game } G_2)| \leq L \cdot \text{Adv}_{k+\ell,k,\eta}^{\text{M-LWE}}(B). \quad (47)$$

Note that in game G_2 , the value \mathbf{v}_i from the challenge ciphertext is independent of bit b and therefore $\Pr(b = b' \text{ in game } G_2) = 1/2$. Collecting the probabilities in (46) and (47) yields the required bound.

$P_{t,\ell}$ -mKEM. An IND-CPA secure $P_{t,\ell}$ -mPKE can be converted into an IND-CCA secure $P_{t,\ell}$ -mKEM via the (generalized) FO transform described in [17]; therefore, the transformation details are omitted.

7 Conclusion

In this paper, we have investigated the effects of ciphertext packing on M-LWE based KEMs like Kyber. We have also demonstrated that by utilizing packed

ciphertexts, the CER of Kyber can be decreased by over 90%, while still maintaining IND-CCA secure. However, a general challenge with ciphertext packing is that the DFR increases linearly with the number of packed ciphertexts. To address this issue, we introduced a coded version of packed Kyber that reduces the DFR to a negligible level. The DFR analysis can be verified numerically and does not rely on independent assumptions about the decoding noise entries. Our findings suggest that M-LWE based cryptosystems can be significantly enhanced in sizes and communication efficiency through advanced techniques in quantization, ciphertext packing, and coding.

References

1. Alkim, E., Bos, J.W., Ducas, L., Longa, P., Mironov, I., Naehrig, M., Nikolaenko, V., Peikert, C., Raghunathan, A., Stebila, D.: FrodoKEM: Learning With Errors Key Encapsulation. Preliminary Standardization Proposal (2024), <https://frodokem.org/>
2. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum Key Exchange—A New Hope. In: 25th USENIX Security Symposium (USENIX Security 16). pp. 327–343. USENIX Association, Austin, TX (Aug 2016), <https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim>
3. Apple Security Engineering and Architecture: iMessage with PQ3: The new state of the art in quantum-secure messaging at scale. Apple Security Research (Feb 2024), <https://security.apple.com/blog/imessage-pq3/>
4. Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J., Schwabe, P., Seiler, G., Stehlé, D.: Algorithm specifications and supporting documentation (version 3.02). Tech. rep., Submission to the NIST post-quantum project (2021), <https://pq-crystals.org/kyber/resources.shtml>
5. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 353–367 (2018). <https://doi.org/10.1109/EuroSP.2018.00032>
6. Brakerski, Z., Gentry, C., Halevi, S.: Packed Ciphertexts in LWE-Based Homomorphic Encryption. In: Kurosawa, K., Hanaoka, G. (eds.) Public-Key Cryptography – PKC 2013. pp. 1–13. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
7. Brakerski, Z., Vaikuntanathan, V.: Efficient Fully Homomorphic Encryption from (Standard) LWE. In: 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science. pp. 97–106 (2011). <https://doi.org/10.1109/FOCS.2011.12>
8. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: TFHE: Fast Fully Homomorphic Encryption over the Torus. *Journal of Cryptology* **33**, 34–91 (2020). <https://doi.org/10.1007/s00145-019-09319-x>
9. Conway, J., Sloane, N.: Fast quantizing and decoding and algorithms for lattice quantizers and codes. *IEEE Transactions on Information Theory* **28**(2), 227–232 (1982). <https://doi.org/10.1109/TIT.1982.1056484>
10. Conway, J., Sloane, N.: Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice. *IEEE Transactions on Information Theory* **32**(1), 41–50 (1986). <https://doi.org/10.1109/TIT.1986.1057135>

11. Conway, J.H., Sloane, N.J.A.: On the Voronoi Regions of Certain Lattices. *SIAM Journal on Algebraic Discrete Methods* **5**(3), 294–305 (1984). <https://doi.org/10.1137/0605031>, <https://doi.org/10.1137/0605031>
12. Conway, J.H., Sloane, N.J.A.: *Sphere Packings, Lattices, and Groups*. Springer-Verlag, New York, 3 edn. (1999). <https://doi.org/10.1007/978-1-4757-6568-7>
13. D’Anvers, J.P., Tiepelt, M., Vercauteren, F., Verbauwhede, I.: Timing Attacks on Error Correcting Codes in Post-Quantum Schemes. In: *Proceedings of ACM Workshop on Theory of Implementation Security Workshop*. p. 2–9. TIS’19, Association for Computing Machinery, New York, NY, USA (2019). <https://doi.org/10.1145/3338467.3358948>
14. Ducas, L., Schanck, J.: *Security Estimation Scripts for Kyber and Dilithium*. GitHub (2019), <https://github.com/pq-crystals/security-estimates>
15. Fritzmann, T., Pöppelmann, T., Sepulveda, J.: Analysis of error-correcting codes for lattice-based key exchange. In: Cid, C., Jacobson Jr., M.J. (eds.) *Selected Areas in Cryptography – SAC 2018*. pp. 369–390. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-10970-7_17
16. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A Modular Analysis of the Fujisaki-Okamoto Transformation. In: Kalai, Y., Reyzin, L. (eds.) *Theory of Cryptography*. pp. 341–371. Springer International Publishing, Cham (2017)
17. Katsumata, S., Kwiakowski, K., Pintore, F., Prest, T.: Scalable Ciphertext Compression Techniques for Post-quantum KEMs and Their Applications. In: Moriai, S., Wang, H. (eds.) *Advances in Cryptology – ASIACRYPT 2020*. pp. 289–320. Springer International Publishing, Cham (2020)
18. Liu, S., Sakzad, A.: CRYSTALS-Kyber With Lattice Quantizer. In: *2024 IEEE International Symposium on Information Theory (ISIT)*. pp. 2886–2891 (2024). <https://doi.org/10.1109/ISIT57864.2024.10619497>
19. Liu, S., Sakzad, A.: Lattice Codes for CRYSTALS-Kyber. In: *Des. Codes Cryptogr.* (2025), <https://doi.org/10.1007/s10623-025-01640-w>
20. Liu, S., Sakzad, A.: Semi-compressed CRYSTALS-Kyber. In: Liu, J.K., Chen, L., Sun, S.F., Liu, X. (eds.) *Provable and Practical Security*. pp. 65–82. Springer Nature Singapore, Singapore (2025). https://doi.org/10.1007/978-981-96-0957-4_4
21. Lloyd, S.: Least squares quantization in PCM. *IEEE Transactions on Information Theory* **28**(2), 129–137 (1982). <https://doi.org/10.1109/TIT.1982.1056489>
22. Lyu, S., Liu, L., Ling, C., Lai, J., Chen, H.: Lattice Codes for Lattice-Based PKE. In: *Des. Codes Cryptogr.* (2023), <https://doi.org/10.1007/s10623-023-01321-6>
23. Maringer, G., Puchinger, S., Wachter-Zeh, A.: Information- and Coding-Theoretic Analysis of the RLWE/MLWE Channel. *IEEE Transactions on Information Forensics and Security* **18**, 549–564 (2023). <https://doi.org/10.1109/TIFS.2022.3226907>
24. Murphy, S., Player, R.: A central limit approach for ring-LWE noise analysis. *IACR Communications in Cryptology* **1**(2) (2024). <https://doi.org/10.62056/ay76c0kr>
25. National Institute of Standards and Technology: *Module-Lattice-Based Key-Encapsulation Mechanism Standard*. Federal Information Processing Standards Publication (FIPS) NIST FIPS 203. (2023). <https://doi.org/10.6028/NIST.FIP.S.203>
26. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: *Annual international cryptology conference*. pp. 554–571. Springer (2008)

27. van Poppel, A.: Cryptographic decoding of the Leech lattice. Cryptology ePrint Archive, Paper 2016/1050 (2016), <https://eprint.iacr.org/2016/1050>
28. Proakis, J.G.: Digital Communications. McGraw-Hill, New York, 4 edn. (2000)
29. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proc. ACM Symp. Theory Comput. (STOC). pp. 84–93 (2005)
30. Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446 (Aug 2018). <https://doi.org/10.17487/RFC8446>, <https://www.rfc-editor.org/info/rfc8446>
31. Saliba, C., Luzzi, L., Ling, C.: A reconciliation approach to key generation based on Module-LWE. In: 2021 IEEE International Symposium on Information Theory (ISIT). pp. 1636–1641 (2021). <https://doi.org/10.1109/ISIT45174.2021.9517882>
32. Sheran, L., Jolene, T., Ajanthan, A., Kyle, N., Rafael, M., Sotirios, D.: Post-quantum readiness for TLS at Meta. Engineering at Meta (May 2024), <https://engineering.fb.com/2024/05/22/security/post-quantum-readiness-tls-pqr-meta/>
33. Smart, N.P.: Efficient Key Encapsulation to Multiple Parties. In: Blundo, C., Ciamato, S. (eds.) Security in Communication Networks. pp. 208–219. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
34. Vardy, A., Be’ery, Y.: Maximum likelihood decoding of the Leech lattice. IEEE Transactions on Information Theory **39**(4), 1435–1444 (1993). <https://doi.org/10.1109/18.243466>

8 Definitions, Syntaxes, and Security Models for Multi-Recipient PKE and KEM, adapted from [17]

In this Appendix, we first provide definitions of mKEM and mPKE. We then provide a generic transformation from mPKE to mKEM.

8.1 Decomposable Multi-Recipient Public Key Encryption

Definition 19 (Decomposable Multi-Recipient Public Key Encryption). A (single-message) decomposable multi-recipient public key encryption (mPKE) over a message space \mathcal{M} and ciphertext spaces \mathcal{C} and $\mathcal{C}_{\text{single}}$ consists of the following five algorithms $\text{mPKE} = (\text{mSetup}, \text{mGen}, \text{mEnc}, \text{mExt}, \text{mDec})$:

- $\text{mSetup}(1^\kappa) \rightarrow \text{pp}$: The setup algorithm on input the security parameter 1^κ outputs a public parameter pp .
- $\text{mGen}(\text{pp}) \rightarrow (\text{pk}, \text{sk})$: The key generation algorithm on input a public parameter pp outputs a pair of public key and secret key (pk, sk) .
- $\text{mEnc}(\text{pp}, (\text{pk}_i)_{i \in [N]}, \text{M}; r_0, r_1, \dots, r_N) \rightarrow \text{ct} = (\text{ct}_0, (\hat{\text{ct}}_i)_{i \in [N]})$: The (decomposable) encryption algorithm running with randomness (r_0, r_1, \dots, r_N) , splits into a pair of algorithms $(\text{mEnc}^i, \text{mEnc}^d)$:
 - $\text{mEnc}^i(\text{pp}; r_0) \rightarrow \text{ct}_0$: On input a public parameter pp and randomness r_0 , it outputs a (public key Independent) ciphertext ct_0 .

- $\text{mEnc}^d(\text{pp}, \text{pk}_i, \text{M}; r_0, r_i) \rightarrow \widehat{\text{ct}}_i$: On input a public parameter pp , a public key pk_i , a message $\text{M} \in \mathcal{M}$, and randomness (r_0, r_i) , it outputs a (public key Dependent) ciphertext $\widehat{\text{ct}}_i$.
- $\text{mExt}(i, \text{ct}) \rightarrow \text{ct}_i = (\text{ct}_0, \widehat{\text{ct}}_i)$ or \perp : The deterministic extraction algorithm on input an index $i \in \mathbb{N}$ and a (multi-recipient) ciphertext $\text{ct} \in \mathcal{C}$, outputs either a (single-recipient) ciphertext $\text{ct}_i = (\text{ct}_0, \widehat{\text{ct}}_i) \in \mathcal{C}_{\text{single}}$ or a special symbol \perp_{Ext} indicating extraction failure.
- $\text{mDec}(\text{sk}, \text{ct}_i) \rightarrow \text{M}$ or \perp : The deterministic decryption algorithm on input a secret key sk and a ciphertext $\text{ct}_i \in \mathcal{C}_{\text{single}}$, outputs either $\text{M} \in \mathcal{M}$ or a special symbol $\perp \notin \mathcal{M}$.

Definition 20 (Correctness). A mPKE is δ -correct if

$$\delta \geq \mathbb{E} \left[\max_{\text{M} \in \mathcal{M}} \Pr_{r_0, r} \left[\begin{array}{l} \text{ct}_0 \leftarrow \text{mEnc}^i(\text{pp}; r_0), \widehat{\text{ct}} \leftarrow \text{mEnc}^d(\text{pp}, \text{pk}, \text{M}; r_0, r) \\ \text{M} \neq \text{mDec}(\text{sk}, (\text{ct}_0, \widehat{\text{ct}})) \end{array} \right] \right], \quad (48)$$

where the expectation is also taken over $\text{pp} \leftarrow \text{mSetup}(1^\kappa)$ and $(\text{pk}, \text{sk}) \leftarrow \text{mGen}(\text{pp})$.

Definition 21 (γ -Spreadness). Let mPKE be a decomposable multi-recipient PKE with message space \mathcal{M} and ciphertext spaces \mathcal{C} and $\mathcal{C}_{\text{single}}$. For all $\text{pp} \in \text{Setup}(1^\kappa)$, and $(\text{pk}, \text{sk}) \in \text{Gen}(\text{pp})$, define

$$\gamma(\text{pp}, \text{pk}) := -\log_2 \left(\max_{\text{ct} \in \mathcal{C}_{\text{single}}, \text{M} \in \mathcal{M}} \Pr_{r_0, r} \left[\text{ct} = (\text{mEnc}^i(\text{pp}; r_0), \text{mEnc}^d(\text{pp}, \text{pk}, \text{M}; r_0, r)) \right] \right).$$

We call mPKE γ -spread if $\mathbb{E}[\gamma(\text{pp}, \text{pk})] \geq \gamma$, where the expectation is taken over $\text{pp} \leftarrow \text{mSetup}(1^\kappa)$ and $(\text{pk}, \text{sk}) \leftarrow \text{mGen}(\text{pp})$.

Definition 22 (IND-CPA). Let mPKE be a decomposable multi-recipient PKE with message space \mathcal{M} and ciphertext space \mathcal{C} . We define IND-CPA by a game illustrated in 2 and say the (possibly quantum) adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ wins if the game outputs 1. We define the advantage of \mathcal{A} against IND-CPA security of mPKE parameterized by $N \in \mathbb{N}$ as

$$\text{Adv}_{\text{mPKE}, N}^{\text{IND-CPA}}(\mathcal{A}) = |\Pr[\mathcal{A} \text{ wins}] - 1/2|.$$

8.2 Multi-Recipient Key Encapsulation Mechanism

Definition 23 (Multi-Recipient Key Encapsulation Mechanism). A (single-message) multi-recipient key encapsulation mechanism (mKEM) over a key space \mathcal{K} and ciphertext space \mathcal{C} consists of the following five algorithms $\text{mKEM} = (\text{mSetup}, \text{mGen}, \text{mEncaps}, \text{mExt}, \text{mDecaps})$:

- $\text{mSetup}(1^\kappa) \rightarrow \text{pp}$: The setup algorithm on input the security parameter 1^κ outputs a public parameter pp .
- $\text{mGen}(\text{pp}) \rightarrow (\text{pk}, \text{sk})$: The key generation algorithm on input a public parameter pp outputs a pair of public key and secret key (pk, sk) .
- $\text{mEncaps}(\text{pp}, (\text{pk}_i)_{i \in [N]}) \rightarrow (\text{K}, \text{ct})$: The encapsulation algorithm on input a public parameter pp , and N public keys $(\text{pk}_i)_{i \in [N]}$, outputs a key K and a ciphertext ct .

GAME IND-CPA

```

1: pp ← mSetup(1κ)
2: for i ∈ [N] do
3:   (pki, ski) ← mGen(pp)
4: end for
5: (M0*, M1*, state) ← A1(pp, (pki)i∈[N])
6: b ← {0, 1}
7: ct* ← mEnc(pp, (pki)i∈[N], Mb*)
8: b' ← A2(pp, (pki)i∈[N], ct*, state)
9: return [b = b']

```

GAME IND-CCA

```

1: pp ← mSetup(1κ)
2: for i ∈ [N] do
3:   (pki, ski) ← mGen(pp)
4: end for
5: (K0*, ct*) ← mEncaps(pp, (pki)i∈[N])
6: K1* ← K
7: b ← {0, 1}
8: b' ← AD(pp, (pki)i∈[N], ct*, Kb*)
9: return [b = b']

```

Decapsulation Oracle D(i, ct)

```

1: cti* := mExt(i, ct)
2: if ct = cti* then
3:   return ⊥
4: end if
5: K := mDecaps(ski, ct)
6: return K

```

Fig. 2. IND-CPA of mPKE and IND-CCA of mKEM.

- $\text{mExt}(i, \text{ct}) \rightarrow \text{ct}_i$: The deterministic extraction algorithm on input an index $i \in \mathbb{N}$ and a ciphertext ct , outputs either ct_i or a special symbol \perp_{Ext} indicating extraction failure.
- $\text{mDecaps}(\text{sk}, \text{ct}_i) \rightarrow \text{K}$ or \perp : The deterministic decryption algorithm on input a secret key sk and a ciphertext ct_i , outputs either $\text{K} \in \mathcal{K}$ or a special symbol $\perp \notin \mathcal{K}$.

Definition 24 (Correctness). A mKEM is δ_N -correct if

$$\delta_N \geq \Pr \left[\begin{array}{l} (\text{K}, \text{ct}) \leftarrow \text{mEnc}(\text{pp}, (\text{pk}_i)_{i \in [N]}), (\text{ct}_i \leftarrow \text{mExt}(i, \text{ct}))_{i \in [N]}; \\ \exists i \in [N] \text{ s.t. } \text{K} \neq \text{mDec}(\text{sk}, \text{ct}_i) \end{array} \right],$$

where the probability is also taken over $\text{pp} \leftarrow \text{mSetup}$ and $(\text{pk}_i, \text{sk}_i) \leftarrow \text{mGen}(\text{pp})$ for all $i \in [N]$.

Definition 25 (IND-CCA). Let mKEM be a multi-recipient KEM. We define IND-CCA by a game illustrated in 2 and say the (possibly quantum) adversary \mathcal{A} (making only classical decapsulation queries to \mathcal{D}) wins if the game outputs 1. We define the advantage of \mathcal{A} against IND-CCA security of mKEM parameterized by $N \in \mathbb{N}$ as

$$\text{Adv}_{\text{mKEM}, N}^{\text{IND-CCA}}(\mathcal{A}) = |\Pr[\mathcal{A} \text{ wins}] - 1/2|.$$

Generic Construction via FO Transform The authors of [17] provided a generic transformation of an IND-CPA secure mPKE to an IND-CCA secure mKEM using a generalized Fujisaki-Okamoto transform, see Fig. 3.

$\frac{\text{mSetup}(1^\kappa)}{1: \text{pp} \leftarrow \text{mSetup}^{\text{P}}(1^\kappa)$ $2: \text{return pp}$	$\frac{\text{mGen}(\text{pp})}{1: (\text{pk}, \text{sk}^{\text{P}}) \leftarrow \text{mGen}^{\text{P}}(\text{pp})}$ $2: \text{seed} \leftarrow \{0, 1\}^\ell$ $3: \text{sk} := (\text{sk}^{\text{P}}, \text{seed})$ $4: \text{return} (\text{pk}, \text{sk})$	$\frac{\text{mExt}(i, \text{ct})}{1: \text{ct}_i \leftarrow \text{mExt}^{\text{P}}(i, \text{ct})}$ $2: \text{return ct}_i$
$\frac{\text{mEncaps}(\text{pp}, (\text{pk}_i)_{i \in [N]})}{1: \text{M} \leftarrow \mathcal{M}$ $2: \text{ct}_0 := \text{mEnc}^i(\text{pp}; \text{G}_1(\text{M}))$ $3: \text{for } i \in [N] \text{ do}$ $4: \quad \widehat{\text{ct}}_i := \text{mEnc}^{\text{d}}(\text{pp}, \text{pk}_i, \text{M}; \text{G}_1(\text{M}), \text{G}_2(\text{pk}_i, \text{M}))$ $5: \text{end for}$ $6: \text{K} := \text{H}(\text{M})$ $7: \text{return} (\text{K}, \text{ct} := (\text{ct}_0, (\widehat{\text{ct}}_i)_{i \in [N]}))$	$\frac{\text{mDecaps}(\text{sk}, \text{ct})}{1: \text{sk} := (\text{sk}^{\text{P}}, \text{seed})}$ $2: \text{M} := \text{mDec}(\text{sk}^{\text{P}}, \text{ct})$ $3: \text{if } \text{M} = \perp \text{ then}$ $4: \quad \text{return K} := \text{H}'(\text{seed}, \text{ct})$ $5: \text{end if}$ $6: \text{ct}_0 := \text{mEnc}^i(\text{pp}; \text{G}_1(\text{M}))$ $7: \widehat{\text{ct}} := \text{mEnc}^{\text{d}}(\text{pp}, \text{pk}, \text{M}; \text{G}_1(\text{M}), \text{G}_2(\text{pk}, \text{M}))$ $8: \text{if } \text{ct} \neq (\text{ct}_0, \widehat{\text{ct}}) \text{ then}$ $9: \quad \text{return K} := \text{H}'(\text{seed}, \text{ct})$ $10: \text{else}$ $11: \quad \text{return K} := \text{H}(\text{M})$ $12: \text{end if}$	

Fig. 3. An IND-CCA secure mKEM from a decomposable IND-CPA secure mPKE = (mSetup^P, mGen^P, mEnc = (mEncⁱ, mEnc^d), mExt^P, mDec). We include the superscript ^P to make the code more readable.

Theorem 1 (IND-CPA mPKE \Rightarrow IND-CCA mKEM, adapted from [17]). *Assume mPKE with message space \mathcal{M} is δ -correct and γ -spread. Then, for any classical PPT IND-CCA adversary \mathcal{A} issuing at most $q_{\mathcal{D}}$ queries to the decapsulation oracle \mathcal{D} , a total of at most q_{G} queries to G_1 and G_2 , and at most q_{H} and q'_{H} queries to H and H' , there exists a classical PPT adversary \mathcal{B}_{IND} such that*

$$\text{Adv}_{\text{mKEM}, N}^{\text{IND-CCA}}(\mathcal{A}) \leq 2 \cdot \text{Adv}_{\text{mPKE}, N}^{\text{IND-CPA}}(\mathcal{B}_{\text{IND}}) + (2q_{\text{G}} + q_{\mathcal{D}} + 2) \cdot \delta$$

$$+ q_{\mathcal{D}} \cdot 2^{-\gamma} + \frac{(q_{\text{G}} + q_{\text{H}})}{|\mathcal{M}|} + q'_{\text{H}} \cdot N \cdot 2^{-\ell}.$$

where the running time of \mathcal{B}_{IND} is about that of \mathcal{A} , and ℓ is bit-length of the seed included in the private key.

Theorem 2 (IND-CPA mPKE \Rightarrow IND-CCA mKEM, adapted from [17]). *Assume mPKE with message space \mathcal{M} is δ -correct and γ -spread. Then, for any quantum PT IND-CCA adversary \mathcal{A} issuing at most $q_{\mathcal{D}}$ classical queries to the decapsulation oracle \mathcal{D} , a total of at most q_{G} quantum queries to G_1 and G_2 , and at most q_{H} and q'_{H} quantum queries to H and H' , there exists a quantum PT adversary \mathcal{B}_{IND} such that*

$$\text{Adv}_{\text{mKEM}, N}^{\text{IND-CCA}}(\mathcal{A}) \leq \sqrt{8 \cdot (q_{\text{G}} + 1) \cdot \text{Adv}_{\text{mPKE}, N}^{\text{IND-CPA}}(\mathcal{B}_{\text{IND}})} + \frac{8 \cdot (q_{\text{G}} + 1)}{\sqrt{|\mathcal{M}|}}$$

$$+ 12 \cdot (q_{\text{G}} + q_{\mathcal{D}} + 1)^2 \cdot \delta_N + q_{\mathcal{D}} \cdot 9\sqrt{2^{-\gamma}} + 9 \cdot 2^{-\frac{\mu}{2}} + q'_{\text{H}} \cdot N \cdot 2^{-\frac{\ell+1}{2}},$$

where the running time of \mathcal{B}_{IND} is about that of \mathcal{A} , ℓ is bit-length of the seed included in the private key, and $\mu = |r_0| + |r|$ for $(r_0, r) \in \mathcal{R}$ where \mathcal{R} is the randomness space of mPKE for a single ciphertext.