Practical Deniable Post-Quantum X3DH: A Lightweight Split-KEM for K-Waay

Guilhem Niot[1,2]

[1]PQShield
[2]Univ Rennes, CNRS, IRISA

This is the full version of a paper to appear in the proceedings of the 20th ACM ASIA Conference on Computer and Communications Security (ACM ASIACCS 2025).

Abstract

The Signal Protocol, underpinning secure messaging for billions, faces the challenge of migrating to a post-quantum world while preserving critical properties like asynchrony and deniability. While Signal's PQXDH key exchange protocol addresses post-quantum confidentiality, full migration of the X3DH protocol remains elusive. Relying on a split KEM (K-Waay, USENIX '24) offers a promising migration path, but it has so far suffered from size limitations compared to concurrent works leveraging ring signatures.

This work introduces Sparrow-KEM and Sym-Sparrow-KEM, novel asymmetric and symmetric split KEMs respectively, i.e. for which keys can be used interchangeably for sending and receiving, or only in one direction. They are designed to optimize the K-Waay protocol for size efficiency. Leveraging the MLWE assumption, these constructions reduce by a factor 5.1× the communication of prior post-quantum X3DH based on split KEMs, plus provides a 40× speedup. Additionally, Sym-Sparrow-KEM is the first symmetric split-KEM to offer deniability, IND-1KCA, and IND-1BatchCCA security, capturing implicit authentication properties. We provide formal security proofs for both schemes, including deniability. Our results demonstrate the feasibility of a compact and deniable post-quantum X3DH protocol based on split KEMs.

1 Introduction

At the heart of many messaging platforms like Signal or Whatsapp lies the X3DH (Triple Diffie-Hellman) key exchange protocol [23], which enables two parties to establish a shared secret key for encrypted communication, even in asynchronous settings where one party might be offline. However, the security of X3DH hinges on the computational difficulty of the Diffie-Hellman problem, which becomes vulnerable with the advent of powerful quantum computers.

Recognizing this critical threat, researchers in academia and the industry (Signal, Apple, etc.) have actively explored post-quantum secure alternatives for X3DH. In 2023, Signal deployed their PQXDH solution [19], an initial post-quantum *hybrid* key exchange protocol. This first step importantly provides post-quantum *confidentiality* and prevents "store-now-decrypt-later" attacks. However, post-quantum *authentication* remains unresolved, and an active quantum attacker would be able to perform impersonation attacks. It seems desirable to design an alternative to X3DH that is fully post-quantum secure.

Standard X3DH revolves around the interplay of long-term and ephemeral key pairs. Parties initially upload to a central server a set of short-lived keys in a so-called prekey bundle, authenticated using their long-term keys. Subsequently, they can initiate communication and derive session keys by downloading a counterpart's bundle and executing multiple Diffie-Hellman key exchanges using a mix of these key types. This intricate design guarantees confidentiality even if ephemeral or long-term keys are compromised.

Additionally, X3DH offers a degree of deniability, as formalized by Vatandas et al. [31], due to its implicit authentication, relying on the Diffie-Hellman key exchanges rather than explicit signatures. Essentially, a

participant can plausibly deny having engaged in a key exchange with another party, providing a crucial privacy safeguard in scenarios like whistleblowing or activism.

Two main approaches have emerged to migrate X3DH to a post-quantum setting while preserving its essential properties: ring signatures and split key encapsulation mechanisms (split-KEMs). Ring signatures offer a promising path by enabling a group member to anonymously sign a message on behalf of the entire group. In the context of X3DH, this could be used to authenticate prekey bundles without revealing the specific signer as studied in [15, 16, 6]. Alternatively, split-KEMs provide a different paradigm for key exchange, where two parties can directly exchange a key, akin to Diffie-Hellman, allowing a quite natural update of X3DH. In a split KEM, a sending party B encapsulates to a counterpart A by using their own secret key sk_B and the public key pk_A to produce a ciphertext ct. A then decapsulates ct using sk_A and pk_B.

Split KEMs were introduced by Brendel et al. [7], who define several levels of security depending on the adversary's access to encapsulation and decapsulation oracles, as a tool akin to Diffie-Hellman for implementing Authenticated Key Exchange (AKE) protocols. They notably identified the conception of a symmetric primitive, i.e. where the key can be used both for sending and receiving, as a desirable property to implement secure messaging. [7, 14, 3] proposed several split KEM constructions achieving from the weakest to the highest level of security defined in [7]. Collins et al. [8] however showed that these security notions are not sufficient for use in Signal's X3DH, as they do not guarantee implicit authentication and prevention of man-in-the-middle attacks. They formally defined additional security notions that enabled them to introduce K-Waay, a post-quantum variant of the X3DH protocol relying on deniable split KEMs. They rely on intermediate notions of security, coined IND-1KCA and IND-1BatchCCA, allowing only one encapsulation, and one decapsulation query (but for many ciphertexts at once). Building on Frodo KEM [5], they introduced FrodoKex+, an asymetric split KEM provably plugable in K-Waay.

While K-Waay presents a significant step forward, and notably provides an improved computational cost over previous works implementing a post-quantum X3DH, its reliance on the Frodo KEM [5] to build a split KEM leads to relatively large sizes (pre-keys and ciphertexts) compared to ring signature-based constructions. This can be a major hurdle for practical adoption, especially in bandwidth-constrained environments like those encountered on mobile devices. Reducing the size of split KEM becomes crucial for seamless integration of post-quantum security into existing messaging protocols like Signal.

Note that deniability is a complex notion in cryptography, influenced by adversarial models and the information available to a judge. Various models are used depending on whether parties behave maliciously or semi-honestly in the protocol, and which information is leaked to the judge. Both HKKP [16] and K-Waay [8] notably considers semi-honest parties for deniability, but HKKP does not cover any secret material leakage, while K-Waay allows leakage of the receiver's secret key.

From previous works, it remains an open question whether K-Waay can achieve the same level of compactness as X3DH protocols based on ring signatures, while keeping its low computational cost advantage.

1.1 Our contributions

In this work, we present novel split-KEM constructions based on MLWE, that significantly improve over previous constructions in both size and efficiency. We integrate them in K-Waay to show that split KEMs can be used to design a post-quantum deniable X3DH protocol as compact as using ring signatures, and with a lower computational cost. We also introduce the first symmetric split KEM pluggable in K-Waay and discuss the implied improvements to the practical deniability properties of the K-Waay protocol.

Our specific contributions include:

1. **Design of novel split-KEM constructions.** We design Sparrow-KEM the first split KEM relying on MLWE that satisfies the IND-1KCA and IND-1BatchCCA security properties, capturing implicit authentication and required for integration in the K-Waay protocol. We also introduce a novel ciphertext truncation technique which allows us to instantiate the first symmetric split KEM pluggable in K-Waay, that we coin Sym-Sparrow-KEM. Its symmetry allows one to decide at a later stage whether it shall be used for sending or receiving messages.

2. **New proof techniques for split KEM deniability.** We prove the deniability of Sparrow-KEM and

Sym-Sparrow-KEM relying on the recent problem Hint-MLWE, that reduces to MLWE, and answers a question left open in K-Waay [8] on how to prove deniability in the module setting.

3. **Improved deniability of K-Waay.** We discuss the improved practical deniability of K-Waay when using a symmetric split KEM, and show that at a small cost, our contributions allow one to use K-Waay with a stronger deniability model and increase the confidence in it.

4. **A compact split KEM, competitive with ring signatures.** We perform an extensive security and parameter analysis for a tight instantiation of our split KEM. Combined with our optimized design, this allows us to shrink the public key size of our split KEM by a factor 10.5× compared to the prior state-of-the-art split KEM from K-Waay [8]. When considering the complete transcript of the K-Waay protocol, we also reduce the communication cost by a factor 5.1×! This closes the gap in size between the two approaches considered to migrate X3DH to the post-quantum setting either based on ring signatures or on split KEMs, and our split KEM-based instantiation has competitive sizes with concurrent approaches relying on ring signatures.

5. **The most efficient post-quantum X3DH protocol.** We provide an implementation for our split KEM Sparrow-KEM. Our benchmarks show that Sparrow-KEM is 40× faster than the split KEM FrodoKex+, which was already the computationally fastest option for a post-quantum X3DH and several times faster than the ring signature approach. Our optimized construction hence increases the gap in terms of efficiency between split KEMs and ring signatures, and shows that split KEMs are strong contenders, especially considering that these algorithms are typically run on user devices, where battery saving is crucial.

2 Preliminaries

We begin by introducing the notations we use throughout this paper. Since this work aims at providing optimized split KEM constructions for K-Waay, we also recall definitions and security notions from K-Waay [8], as well as hardness assumptions we will rely on.

Note that for space reasons we only recall the K-Waay protocol at a high-level and the abstracted properties it requires from a split-KEM to be instanciated.

2.1 Notations

We note κ the security parameter.

Sets, functions and distributions. For an integer $N > 0$, we note $[N] = \{0, \dots, N-1\}$. To denote the assign operation, we use $y \coloneqq f(x)$ when f is deterministic and $y \leftarrow f(x)$ when randomized. When S is a finite set, we note $\mathcal{U}(S)$ the uniform distribution over S, and shorthand $x \xleftarrow{\$} S$ for $x \leftarrow \mathcal{U}(S)$.

Linear algebra. Throughout the work, for a fixed power-of-two n, we note $\mathcal{K} = \mathbb{Q}[x]/(x^n + 1)$ and $\mathcal{R} = \mathbb{Z}[x]/(x^n + 1)$ the associated cyclotomic field and cyclotomic ring. We also note $\mathcal{R}_q = \mathcal{R}/(q\mathcal{R})$. Given $\mathbf{x} \in \mathcal{K}^\ell$, we abusively note $\|\mathbf{x}\|$ the Euclidean norm of the $(n\,\ell)$-dimensional vector of the coefficients of $\mathbf{x}$. By default, vectors are treated as *column* vectors (unless specified otherwise).

2.2 Distributions

Definition 1 (Discrete Gaussians). *Given a positive σ, we note σ the Gaussian function defined over $\mathbb{R}^m$ as*

$$\rho_\sigma(\mathbf{x}) = \exp\left(-\|\mathbf{x}\|^2/(2\sigma^2)\right)$$

The above definition may be extended to $\mathbf{x} \in \mathcal{K}^m$ by treating $\mathbf{x}$ as a vector in $\mathbb{R}^{nm}$.

We may note $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \rho_\sigma(\mathbf{x} - \mathbf{c})$.

For any countable set $S \subset \mathcal{K}^m$, we note $\rho_{\sigma,\mathbf{c}}(S) = \sum_{\mathbf{x}\in\mathcal{K}^m} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$ whenever this sum converges. Finally, when $\rho_{\sigma,\mathbf{c}}(S)$ converges, the discrete Gaussian distribution $D_{S,\mathbf{c},\sigma}$ is defined over S by its probability distribution function:

$$D_{S,\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\rho_{\sigma,\mathbf{c}}(S)}. \tag{1}$$

We may omit c when it is equal 0, and S when it is clear from context, and write $D_\sigma \coloneqq D_{S,\sigma,0}$.

Definition 2 (Centered binomial). *We define the centered binomial distribution* Bin *as* $\mathsf{Bin}(1) = \mathsf{Bin}(-1) = 1/4$, $\mathsf{Bin}(0) = 1/2$.

2.3 Split Key Encapsulation Mechanism (Split KEM)

The primitive at the core of this work is split-KEM. It was introduced by Brendel et al. in [7] as a practical tool to provide post-quantum security in X3DH. K-Waay [8] extended their work by introducing an asymmetric split KEM, which allows the exchange of a key between two parties, where one is typically the sender, while the other is the receiver. They also provide a generic construction to achieve a weak variant of IND-CCA security, and deniability properties within their proposal of a post-quantum X3DH protocol. We recall definitions and security notions from [8].

For clarity of the notations, we consider that Alice "A" takes the role of the receiver, and Bob "B" takes the role of the sender in the asymmetric split KEM.

Definition 3 (Split-KEM [7, 8]). *An (asymmetric) split-KEM* sKEM *is a tuple of four efficient algorithms* $(\mathsf{KeygenA}, \mathsf{KeygenB}, \mathsf{Encaps}, \mathsf{Decaps})$ *defined as follows:*

- $(\mathsf{pk_A}, \mathsf{sk_A}) \leftarrow \mathsf{KeygenA}(1^\lambda)$*: The key generation function of the receiving party takes the security parameter λ as input, and outputs a pair of public/secret keys* $(\mathsf{pk_A}, \mathsf{sk_A})$.
- $(\mathsf{pk_B}, \mathsf{sk_B}) \leftarrow \mathsf{KeygenB}(1^\lambda)$*: The key generation function of the sending party takes the security parameter λ as input, and outputs a pair of public/secret keys* $(\mathsf{pk_B}, \mathsf{sk_B})$.
- $K, \mathsf{ct} \leftarrow \mathsf{Encaps}(\mathsf{pk_A}, \mathsf{sk_B})$*: The encapsulation function takes the public key* $\mathsf{pk_A}$ *of a party* A *and the other party's secret key* $\mathsf{sk_B}$ *as inputs, and outputs a ciphertext* ct *and a key K.*
- $K/\perp \leftarrow \mathsf{Decaps}(\mathsf{pk_B}, \mathsf{sk_A}, \mathsf{ct})$*: The decapsulation function takes the secret key* $\mathsf{sk_A}$ *of a party* A*, the other party's public key* $\mathsf{pk_B}$ *and a ciphertext* ct *as inputs, and outputs a key K or the error symbol $\perp$.*

A split KEM sKEM *is said symmetric if* $\mathsf{KeygenA} = \mathsf{KeygenB}$.

Definition 4 (split-KEM Correctness). *A split-KEM* sKEM *is δ-correct if*

$$\Pr\left[K \neq K' : \begin{array}{c}(\mathsf{pk_A}, \mathsf{sk_A}) \leftarrow \mathsf{KeygenA}(1^\lambda);\\ (\mathsf{pk_B}, \mathsf{sk_B}) \leftarrow \mathsf{KeygenB}(1^\lambda);\\ K, \mathsf{ct} \leftarrow \mathsf{Encaps}(\mathsf{pk_A}, \mathsf{sk_B});\\ K' \leftarrow \mathsf{Decaps}(\mathsf{pk_B}, \mathsf{sk_A}, \mathsf{ct})\end{array}\right] \leqslant \delta$$

Split KEMs were introduced as a mean to allow an encapsulating party to contribute to the shared key, while normal KEMs only allow the shared key to be derived from the receiver's key. Their main advantage over normal KEMs is the possibility to implicitly authenticate a sender, which is of great use within a key exchange protocol.

The original work on Split-KEMs [7] introduced the security notion of mm-IND-CCA, which captures the secrecy of a shared secret between two parties. However, Collins et al. [8] explained that this notion is not sufficient to capture implicit authentication, and to prevent impersonation. They introduced the notion of UNF-1KCA security, ensuring that decapsulation will only succeed if encapsulation was performed by the expected party.

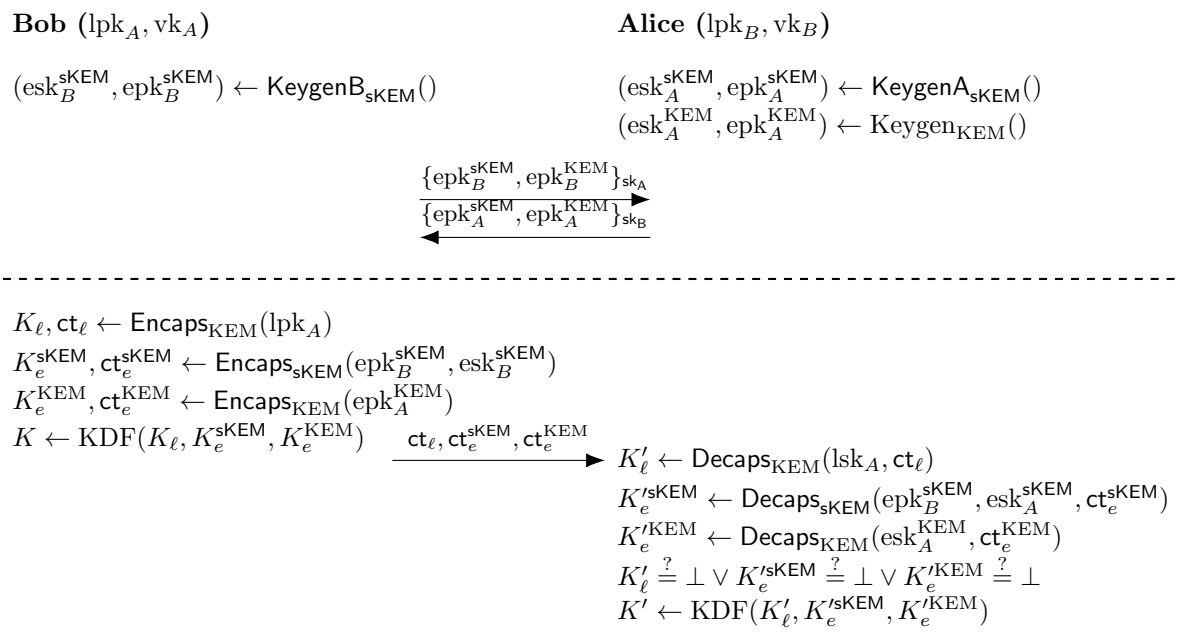

Figure 1: High-level overview of the K-Waay key-exchange protocol (without batching). The communication above the dashed line corresponds to the generation and transfers of pre-key bundles. Values in brackets $\{\cdot\}_{\mathsf{sk}}$ are signed with sk and the signature is verified upon reception. Each party possesses a long-term KEM public key lpk and signature key vk. We use a key-derivation function KDF. In the finalized protocol, Alice batches decapsulations and aborts if any partial decapsulation fails.

2.4 The K-Waay key exchange protocol

K-Waay builds a secure key exchange protocol with a Split KEM relying on the implicit authentication property UNF-1KCA, and on a weakened notion of IND-CCA, coined IND-1BatchCCA, allowing only one batched decapsulation query. In particular, if any of d ciphertexts fails to decapsulate, the entire batched decapsulation aborts. This allows one to limit the leakage from a failure to a single information bit, instead of d bits. Remark that UNF-1KCA (resp. IND-1BatchCCA) only allows one encapsulation query (resp. decapsulation query), which is sufficient for K-Waay where keys are used only once. Designing a sKEM simultaneously achieving both security notions with an unlimited number of queries remains an open question. We recall the high-level construction of K-Waay in Fig. 1, and property definitions below.

Definition 5 (split-KEM UNF-1KCA and IND-1BatchCCA). *We consider the* UNF-1KCA *and* IND-1BatchCCA *games defined in Figure 2. A split-KEM scheme* sKEM *is* UNF-1KCA *and* IND-1BatchCCA *if for any efficient adversary $\mathcal{A}$ we have*

$$\mathsf{Adv}^{\text{unf-1kca}}_{\mathsf{sKEM}}(\mathcal{A}) := \left|\Pr\left[\text{UNF-1KCA}_{\mathsf{sKEM}}(\mathcal{A}) \Rightarrow 1\right] - \frac{1}{2}\right| = \mathsf{negl}$$

$$\mathsf{Adv}^{\text{ind-1batchcca}}_{\mathsf{sKEM}}(\mathcal{A}) := \left|\Pr\left[\text{IND-1BatchCCA}_{\mathsf{sKEM}}(\mathcal{A}) \Rightarrow 1\right] - \frac{1}{2}\right| = \mathsf{negl}$$

Analogously to generic transforms from IND-CPA to IND-CCA for normal KEMs, Collins et al. showed that it suffices to build a Split-KEM verifying two weaker properties coined OW-CPA and decaps-OW-CPA. They propose an efficient transform that then turns such a Split KEM to be UNF-1KCA and IND-1BatchCCA secure. We recall definitions of this weaker security notions, as well as the generic split KEM transform in Fig. 4.

$\text{IND-1BatchCCA}_{\mathsf{sKEM}}(\mathcal{A})$

$b \xleftarrow{\$} \{0,1\}; q := 0$
$\mathsf{pk_A}, \mathsf{sk_A} \leftarrow \mathsf{KeygenA}(1^\lambda)$
$\mathsf{pk_B}, \mathsf{sk_B} \leftarrow \mathsf{KeygenB}(1^\lambda)$
$K_0, \mathsf{ct}^* \leftarrow \mathsf{Encaps}(\mathsf{pk_A}, \mathsf{sk_B})$
$K_1 \xleftarrow{\$} \mathcal{K}$
$b' \leftarrow \mathcal{A}^{\text{BatchDec}}(\mathsf{pk_A}, \mathsf{pk_B}, \mathsf{ct}^*, K_b)$
return $1_{b=b'}$

$\text{BatchDec}(\{(\mathsf{pk}_i, \mathsf{ct}_i)\}_{i\in[d]})$

if $q = 1$ **then return** $\perp$
else $q = 1$
for $i \in [d]$ **do**
  if $(\mathsf{pk}_i, \mathsf{ct}_i) = (\mathsf{pk}_B, \mathsf{ct}^*)$ **then**
    return $\perp$
  $K'_i \leftarrow \mathsf{Decaps}(\mathsf{pk}_i, \mathsf{sk_A}, \mathsf{ct}_i)$
if $\exists i, K'_i = \perp$ **then return** $\perp$
return $(K'_i)_{i\in[d]}$

$\text{UNF-1KCA}_{\mathsf{sKEM}}(\mathcal{A})$

$\mathsf{pk_A}, \mathsf{sk_A} \leftarrow \mathsf{KeygenA}(1^\lambda)$
$\mathsf{pk_B}, \mathsf{sk_B} \leftarrow \mathsf{KeygenB}(1^\lambda)$
$\mathsf{st}, \mathsf{pk} \leftarrow \mathcal{A}(\mathsf{pk_A}, \mathsf{pk_B})$
$K_\mathsf{B}, \mathsf{ct} \leftarrow \mathsf{Encaps}(\mathsf{pk}, \mathsf{sk_B})$
$\mathsf{ct}' \leftarrow \mathcal{A}(\mathsf{st}, \mathsf{pk_B}, \mathsf{sk_A}, \mathsf{ct})$
if $\mathsf{ct}' = \mathsf{ct}$ **then return** 0;
$K_\mathsf{A} \leftarrow \mathsf{Decaps}(\mathsf{pk_B}, \mathsf{sk_A}, \mathsf{ct}')$
return $K_\mathsf{A} \neq \perp$

Figure 2: IND-1BatchCCA and UNF-1KCA games.

$\text{OW-CPA}_{\mathsf{sKEM}}(\mathcal{A})$

$\mathsf{pk_A}, \mathsf{sk_A} \leftarrow \mathsf{KeygenA}(1^\lambda)$
$\mathsf{pk_B}, \mathsf{sk_B} \leftarrow \mathsf{KeygenB}(1^\lambda)$
$K^*, \mathsf{ct}^* \leftarrow \mathsf{Encaps}(\mathsf{pk_A}, \mathsf{sk_B})$
$K' \leftarrow \mathcal{A}(\mathsf{pk_A}, \mathsf{pk_B}, \mathsf{ct}^*)$
return $K' = K^*$

$\text{decaps-OW-CPA}_{\mathsf{sKEM}}(\mathcal{A})$

$\mathsf{pk_A}, \mathsf{sk_A} \leftarrow \mathsf{KeygenA}(1^\lambda)$
$\mathsf{pk_B}, \mathsf{sk_B} \leftarrow \mathsf{KeygenB}(1^\lambda)$
$\mathsf{st}, \mathsf{pk} \leftarrow \mathcal{A}(\mathsf{pk_A}, \mathsf{pk_B})$
$K_\mathsf{B}, \mathsf{ct} \leftarrow \mathsf{Encaps}(\mathsf{pk}, \mathsf{sk_B})$
$K'_\mathsf{A}, \mathsf{ct}' \leftarrow \mathcal{A}(\mathsf{st}, \mathsf{pk_A}, \mathsf{pk_B}, \mathsf{ct})$
$K_\mathsf{A} \leftarrow \mathsf{Decaps}(\mathsf{pk_B}, \mathsf{sk_A}, \mathsf{ct}')$
if $K_\mathsf{A} = \perp$ **then**
  return 0
return $K_\mathsf{A} = K'_\mathsf{A}$

Figure 3: OW-CPA and decaps-OW-CPA games.

Definition 6 (split-KEM OW-CPA). *We consider the* OW-CPA *game defined in Figure 3. A split-KEM scheme* sKEM *is* OW-CPA *if for any efficient adversary* $\mathcal{A}$ *we have*

$$\mathsf{Adv}^{\text{ow-cpa}}_{\mathsf{sKEM}}(\mathcal{A}) := \left|\Pr\left[\text{OW-CPA}_{\mathsf{sKEM}}(\mathcal{A}) \Rightarrow 1\right] - \frac{1}{2}\right| = \mathsf{negl}$$

Definition 7 (split-KEM decaps-OW-CPA). *We consider the* decaps-OW-CPA *game defined in Figure 3. A split-KEM scheme* sKEM *is* decaps-OW-CPA *if for any efficient adversary* $\mathcal{A}$ *we have*

$$\mathsf{Adv}^{\text{decaps-ow-cpa}}_{\mathsf{sKEM}}(\mathcal{A}) := \left|\Pr\left[\text{decaps-OW-CPA}_{\mathsf{sKEM}}(\mathcal{A}) \Rightarrow 1\right] - \frac{1}{2}\right| = \mathsf{negl}$$

Theorem 1 (Theorems 5 and 6 from [8]). *Let* sKEM *be any split-KEM, and* sKEM*' be the split-KEM transform from Fig. 4 applied to* sKEM*. In the random oracle model (ROM),* sKEM*' is UNF-1KCA and IND-1BatchCCA if* sKEM *is* OW-CPA *and* Decaps-OW-CPA*. Formally, if* H, H' *are modeled as random oracles with output size* s*, and queried up to* $Q_H, Q_{H'}$ *times, we have:*

$$\mathsf{Adv}^{\text{unf-1kca}}_{\mathsf{sKEM}'}(\mathcal{A}) \leqslant \frac{Q_{H'}^2 + 1}{2^s} + (Q_H + Q_{H'} + 1) \cdot \mathsf{Adv}^{\text{decaps-ow-cpa}}_{\mathsf{sKEM}}(\mathcal{B})$$

$$\mathsf{Adv}^{\text{ind-1batchcca}}_{\mathsf{sKEM}'}(\mathcal{A}) \leqslant \frac{Q_{H'}^2 + d}{2^s} + 2(Q_H + Q_{H'} + d) \cdot \mathsf{Adv}^{\text{ow-cpa}}_{\mathsf{sKEM}}(\mathcal{C})$$

$\mathsf{KeygenA} = \mathsf{sKEM.KeygenA}$ // unchanged key generation
$\mathsf{KeygenB} = \mathsf{sKEM.KeygenB}$

$\mathsf{Encaps}(\mathsf{pk_A}, \mathsf{sk_B})$

1: $K_0, \mathsf{ct}_0 \leftarrow \mathsf{sKEM.Encaps}(\mathsf{pk_A}, \mathsf{sk_B})$
2: $t := H'(\mathsf{pk_A}, \mathsf{pk_B}, \mathsf{ct}_0, K_0)$
3: $K := H(\mathsf{pk_A}, \mathsf{pk_B}, \mathsf{ct}_0, K_0)$
4: $\mathsf{ct} := (\mathsf{ct}_0, t)$
5: **return** (K, ct)

$\mathsf{Decaps}(\mathsf{pk_B}, \mathsf{sk_A}, \mathsf{ct} = (\mathsf{ct}_0, t))$

1: $K' \leftarrow \mathsf{sKEM.Decaps}(\mathsf{pk_B}, \mathsf{sk_A}, \mathsf{ct}_0)$
2: **if** $t \neq H'(\mathsf{pk_A}, \mathsf{pk_B}, \mathsf{ct}_0, K_0)$ **then**
3: **return** $\perp$
4: $K := H(\mathsf{pk_A}, \mathsf{pk_B}, \mathsf{ct}_0, K_0)$
5: **return** K

Figure 4: Generic split KEM transform [8] from OW-CPA and Decaps-OW-CPA to UNF-1KCA and IND-1BatchCCA. H, H' are modeled as random oracles.

$\mathrm{DENY}^{\mathsf{REAL}}_{\mathsf{sKEM}}(\mathcal{A})$

1: $(\mathsf{pk_A}, \mathsf{sk_A}) \leftarrow \mathsf{KeygenA}(1^\lambda)$
2: $(\mathsf{pk_B}, \mathsf{sk_B}) \leftarrow \mathsf{KeygenB}(1^\lambda)$
3: $K, \mathsf{ct} \leftarrow \mathsf{Encaps}(\mathsf{pk_A}, \mathsf{sk_B})$
4: $b \leftarrow \mathcal{A}(\mathsf{pk_A}, \mathsf{pk_B}, \mathsf{sk_A}, K, \mathsf{ct})$
5: **return** b

$\mathrm{DENY}^{\mathsf{SIM}}_{\mathsf{sKEM},\mathsf{sim}}(\mathcal{A})$

1: $(\mathsf{pk_A}, \mathsf{sk_A}) \leftarrow \mathsf{KeygenA}(1^\lambda)$
2: $(\mathsf{pk_B}, \mathsf{sk_B}) \leftarrow \mathsf{KeygenB}(1^\lambda)$
3: $K, \mathsf{ct} \leftarrow \mathsf{sim}(\mathsf{pk_B}, \mathsf{sk_A})$
4: $b \leftarrow \mathcal{A}(\mathsf{pk_A}, \mathsf{pk_B}, \mathsf{sk_A}, K, \mathsf{ct})$
5: **return** b

Figure 5: Deniability games for split-KEMs.

for efficient adversaries $\mathcal{B}, \mathcal{C}$ and d is the number of tuples submitted to BatchDec in the IND-1BatchCCA game.

Finally, we recall the notion of deniability for split-KEM from K-Waay [8]. Intuitively, a split-KEM is deniable if a ciphertext/key pair generated by the receiver Alice is indistinguishable from a ciphertext/key pair generated by the sender Bob. That implies that, assuming the parties behave honestly, a judge cannot be convinced that Bob actually sent a ciphertext to Alice that decapsulated to K (i.e. Alice could have computed this ciphertext herself).

Different notions of deniability are used throughout the literature, differing in the material that is leaked to the judge. The definition of deniability from [9] is the most generic and is agnostic to the form of leakage. Works such as [15] proved deniability only when secret materials are not leaked. As in the original work on K-Waay [8], we consider a deniability notion with partial leakage only, i.e. where only the secret receiver key $\mathsf{sk_A}$ leaks, and the sender's key remains secret.

Definition 8 (Deniability under leak of receiver's key [8]). *A split-KEM* sKEM *is deniable if there exists an efficient simulator* sim *s.t. for all efficient adversaries $\mathcal{A}$, we have*

$$\mathsf{Adv}^{\mathsf{deny}}_{\mathsf{sKEM},\mathsf{sim}}(\mathcal{A}) := |\Pr[\mathrm{DENY}^{\mathsf{REAL}}_{\mathsf{sKEM}}(\mathcal{A}) \Rightarrow 1] - \Pr[\mathrm{DENY}^{\mathsf{SIM}}_{\mathsf{sKEM},\mathsf{sim}}(\mathcal{A}) \Rightarrow 1]| = \mathsf{negl}$$

where $\mathrm{DENY}^{\mathsf{REAL}}$ *and* $\mathrm{DENY}^{\mathsf{SIM}}$ *are defined in Figure 5.*

2.5 Hardness Assumptions

We recall the Module Learning with Errors assumption.

Definition 9 (MLWE). *Let ℓ, k, q be integers, χ be a probability distributions over $\mathcal{R}_q^{k+\ell}$. The advantage $\mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{A}}$ of an adversary $\mathcal{A}$ against the Module Learning with Errors problem $\mathsf{MLWE}_{k,\ell,q,\chi}$ is defined as:*

$$\left|\Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{A}, \begin{bmatrix}\mathbf{I}_k & \mathbf{A}\end{bmatrix} \cdot \mathbf{s}\right)\right] - \Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{A}, \mathbf{u}\right)\right]\right|,$$

where $(\mathbf{A}, \mathbf{u}) \xleftarrow{\$} \mathcal{R}^{k \times \ell} \times \mathcal{R}_q^k$, $\mathbf{s} \leftarrow \chi$. The $\mathsf{MLWE}_{k,\ell,q,\chi}$ assumption states that any efficient adversary $\mathcal{A}$ has a negligible advantage.

We also recall the Hint-MLWE assumption [18], and extended by Espitau, Niot and Prest [11] to capture hints on linear combinations of the MLWE secret and error, i.e. values of the form $\mathbf{z} \coloneqq \mathbf{c}^\top \cdot \mathbf{s} + \mathbf{y}$ where $\mathbf{y}$ is a masking value. We state here a simplified version of [11, Definition 3] limited to polynomial combinations of the secret and error.

Definition 10 (Hint-MLWE). *Let k, ℓ, q, Q be integers, χ, ξ be probability distributions over $\mathcal{R}^{k+\ell}$ and $\mathcal{R}$ respectively, and $\mathcal{C}$ be a distribution of challenges over $\mathcal{R}^{k+\ell}$. The advantage $\mathsf{Adv}^{\mathsf{Hint\text{-}MLWE}}_{\mathcal{A}}$ of an adversary $\mathcal{A}$ against the Hint Module Learning with Errors problem $\mathsf{Hint\text{-}MLWE}_{k,\ell,q,Q,\chi,\xi,\mathcal{C}}$ is defined as:*

$$\left|\Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{A}, \left[\mathbf{I}_k \ \mathbf{A}\right]\mathbf{s}, (\mathbf{c}_i, \mathbf{z}_i)_{i\in[Q]}\right)\right] - \Pr\left[1 \leftarrow \mathcal{A}\left(\mathbf{A}, \mathbf{u}, (\mathbf{c}_i, \mathbf{z}_i)_{i\in[Q]}\right)\right]\right|$$

where $(\mathbf{A}, \mathbf{u}) \xleftarrow{\$} \mathcal{R}_q^{k\times\ell} \times \mathcal{R}_q^k$, $\mathbf{s} \leftarrow \chi$ and for $i \in [Q]$: $\mathbf{c}_i \leftarrow \mathcal{C}$, $\mathbf{y}_i \leftarrow \xi$, and $\mathbf{z}_i = \mathbf{c}_i^\top \cdot \mathbf{s} + \mathbf{y}_i$ are hints on the secret. The $\mathsf{Hint\text{-}MLWE}_{k,\ell,q,Q,\chi,\xi,\mathcal{C}}$ assumption states that any efficient adversary $\mathcal{A}$ has a negligible advantage.

The spectral norm $s_1(\mathbf{M})$ of a matrix $\mathbf{M}$ is defined as the value $\max_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{M}\,\mathbf{x}\|}{\|\mathbf{x}\|}$. We recall that if a matrix is symmetric, then its spectral norm is also its largest eigenvalue. Given a polynomial $c \in \mathcal{R}$, we may abusively use the term "spectral norm $s_1(c)$ of c" when referring to the spectral norm of the anti-circulant matrix $\mathcal{M}(c)$ associated to c. Finally, if $c(x) = \sum_{0\leqslant i<n} c_i\, x^i$, then the Hermitian adjoint of c, which we denote by c^*, is defined as $c^*(x) = c_0 - \sum_{0<i<n} c_{n-i}\, x^i$. Note that $\mathcal{M}(c)^t = \mathcal{M}(c^*)$.

Theorem 2 (Hardness of Hint-MLWE, adapted from [18] and [11]). *Let $\mathcal{C}$ be a distribution over $\mathcal{R}$, and let B_{HMLWE} be a real number such that $s_1(D) \leqslant B_{\mathsf{HMLWE}}$ with overwhelming probability, where $D = \sum_Q \langle \mathbf{c}_i, \mathbf{c}_i^* \rangle \in \mathcal{R}_q$. Let $\sigma, \sigma_{\mathsf{sk}}, \sigma_{\mathsf{pert}} > 0$ such that $\frac{1}{\sigma^2} = 2\left(\frac{1}{\sigma_{\mathsf{sk}}^2} + \frac{B_{\mathsf{HMLWE}}}{\sigma_{\mathsf{pert}}^2}\right)$. If $\sigma \geqslant \sqrt{2}\eta_\varepsilon(\mathbb{Z}^{n(k+\ell)})$ for $0 < \varepsilon \leqslant 1/2$, where $\eta_\varepsilon(\mathbb{Z}^{n(k+\ell)})$ is the smoothing parameter of $\mathbb{Z}^{n(k+\ell)}$, then the problem $\mathsf{MLWE}_{k,\ell,q,\sigma}$ efficiently reduces to $\mathsf{Hint\text{-}MLWE}_{k,\ell,q,Q,\sigma_{\mathsf{sk}},\sigma_{\mathsf{pert}},\mathcal{C}}$ with an advantage increased by at most 2ε.*

2.6 Lattice toolbox

Rounding functions. Given two parameters q and $B < \log q - 1$, we define the rounding function, and the cross-rounding function $\langle\cdot\rangle_{q,B}$ as follows:

$$\lfloor\cdot\rceil_{q,B} : v \mapsto \left\lfloor \frac{2^B}{q} \cdot v \right\rceil \bmod 2^B, \quad \langle\cdot\rangle_{q,B} : v \mapsto \left\lfloor \frac{2^{B+1}}{q} \cdot v \right\rceil \bmod 2$$

where $\lfloor\cdot\rceil : \mathbb{R} \mapsto \mathbb{Z}$ is the "rounding half-up" $\lfloor x \rceil = \lfloor x + \frac{1}{2} \rfloor$ and half-way values are rounded up: e.g. $\lfloor 2.5 \rceil = 3$ and $\lfloor -2.5 \rceil = -2$.

Reconciliation function. We recall the (generalized) reconciliation mechanism from Bos et al. and Peikert [5, 27], which for every approximate agreement in $\mathbb{Z}_q$ allows extracting shared bits. We refer the reader to the aforementioned works for more details. Let q be a positive integer. Let B be the number of bits we want to extract from one coefficient in $\mathbb{Z}_q$ so that $B < \log q - 1$. Now, for any $v \in \mathbb{Z}_q$, which is represented as an integer in $[0, q)$, we define the following functions.

Definition 11 (Randomised doubling function dbl). *For any $v \in \mathbb{Z}_q$, we define $\mathsf{dbl}(\cdot)$ as follows:*

$$\mathsf{dbl}(v) : v \mapsto 2v - e, \quad e \leftarrow \mathsf{Bin}$$

Now, we are ready to define the reconciliation function $\mathsf{Rec} : \mathbb{Z}_{2q} \times \mathbb{Z}_2 \to \mathbb{Z}_{2^B}$.

Definition 12 (Reconciliation function Rec). *For any $w \in \mathbb{Z}_{2q}$ and bit $b \in \{0, 1\}$, let v be the closest element to $w \in \mathbb{Z}_{2q}$ s.t. $\langle v \rangle_{2q,B} = b$. Then, we define Rec as*

$$\mathsf{Rec}_B(w, b) \coloneqq \lfloor v \rceil_{2q,B}$$

The next result gives an important property of the reconciliation function Rec, as described by Peikert [27, Section 3.2].

Lemma 1. *Let q be odd and $\bar{v} \leftarrow \mathsf{dbl}(v)$. If $|v - w| \leqslant \left\lfloor \frac{q}{2^{B+2}} \right\rfloor$ then*

$$\mathsf{Rec}_B(2w, \langle \bar{v} \rangle_{2q,B}) = \lfloor \bar{v} \rceil_{2q,B}$$

Finally, we define the $\mathsf{HelpRec}_B : \mathbb{Z}_q \mapsto \{0,1\}$ function as follows:

Definition 13 ($\mathsf{HelpRec}$ function). *On any input $v \in \mathbb{Z}_q$,*

$$\mathsf{HelpRec}_B(v) \coloneqq \langle \bar{v} \rangle_{2q,B}, \qquad \textit{where } \bar{v} \leftarrow \mathsf{dbl}(v)$$

All the functions above can be naturally generalized to take as input polynomials in $\mathcal{R}_q$ by applying the function to each of the coefficients.

2.7 Useful lemmas

We additionally recall some useful lemmas.

Definition 14 (Dual of a lattice). *The dual of a lattice L, noted $L^\vee$, is the set $L^\vee = \{\mathbf{x} \in \mathsf{Span}_{\mathbb{R}}(L) \mid \forall \mathbf{v} \in L, \langle x, v \rangle \in \mathbb{Z}\}$.*

Definition 15 (Smoothing parameter [24]). *For any lattice L, and any real $\varepsilon > 0$, the smoothing parameter $\eta_\varepsilon(L)$ is the smallest real $s > 0$ such that $\rho_{1/(\sqrt{2\pi}s)}(L^\vee \backslash \{0\}) \leqslant \varepsilon$. We also define a scaled variant $\eta'_\varepsilon(L) := \frac{1}{\sqrt{2\pi}} \eta_\varepsilon(L)$.*

Lemma 2 (Simplified convolution lemma [26, Thm. 4.5]). *Let positive reals σ_1, σ_2 such that $\sigma_1^{-2} + \sigma_2^{-2} \leqslant \eta'_\varepsilon(\mathbb{Z}^n)^{-2}$ for some positive $\varepsilon \leqslant 1/4$. Then the distribution $D_{\mathbb{Z}^n,\sigma_1} + D_{\mathbb{Z}^n,\sigma_2}$ is within statistical distance 8ε of $D_{\mathbb{Z}^n,\sqrt{\sigma_1^2+\sigma_2^2}}$.*

We additionally provide a useful bound on the spectral norm of Gaussian vectors that was communicated to us by a private communication, and is part of another paper currently under review. We do not claim ownership of the following lemma, but we provide its proof in Section B for completeness.

Lemma 3 (Heuristic spectral norm of gaussian vector, from private communication). *Let $\mathbf{x} \leftarrow \mathcal{D}_{\mathcal{R}^\ell,\sigma}$.*

It holds that $s_1(\mathbf{x}) \leqslant \tau\sigma\sqrt{n\ell}$, except with negligible probability if $\tau \geqslant \sqrt{D+1}$ where $D = \frac{\log n + (\kappa-1)\log 2}{\ell}$.

3 Sparrow-KEM: a lightweight split KEM for K-Waay

This section introduces Sparrow-KEM (and its variant Sym-Sparrow-KEM), our split-KEM construction designed to optimize the K-Waay protocol for compactness while maintaining strong security guarantees.

Sparrow-KEM takes inspiration in the prior lattice-based sKEM constructions from Brendel at al. [7] and K-Waay [8] to build a scheme relying on the hardness of MLWE.

At a high level, both of these constructions sample keys of the form $\mathbf{B}_\mathsf{A} \coloneqq \mathbf{A} \cdot \mathbf{S}_\mathsf{A} + \mathbf{E}_\mathsf{A}$ and $\mathbf{B}_\mathsf{B} \coloneqq \mathbf{A}^\top \cdot \mathbf{S}_\mathsf{B} + \mathbf{E}_\mathsf{B}$ for some $\mathbf{S}_\mathsf{A}, \mathbf{S}_\mathsf{B}, \mathbf{E}_\mathsf{A}, \mathbf{E}_\mathsf{B}$ with small coefficients. The core idea for them to exchange a key lies in realizing that the values $\mathbf{S}_\mathsf{B}^\top \cdot \mathbf{B}_\mathsf{A}$ and $\mathbf{B}_\mathsf{B}^\top \cdot \mathbf{S}_\mathsf{A}$ are at a short distance, and hard to compute for any other party than A and B. Leveraging a reconciliation mechanism such as the one described in Section 2.6, it is possible to extract shared bits from these values.

However, both of these constructions present important limitations. The Split-KEM from [7] is rather compact as it is instantiated from module lattice assumptions, but it is only proven IND-CPA secure, and thus cannot be used within a key-exchange protocol. FrodoKex+ from [8] achieves the stronger property UNF-1KCA ensuring implicit authentication, however it builds on plain lattice assumptions and suffers from large sizes and overall inefficiency.

Sparrow-KEM offers the best of both schemes by providing the compact sizes enabled by MLWE, and the stronger security properties required for a key-exchange protocol.

During a global setup phase formalized in Algorithm 1, a shared public matrix $\mathbf{A} \in \mathcal{R}_q^{k \times \ell}$ is sampled. Then, the receiver A samples its public key as $\mathbf{b}_\mathsf{A} := \mathbf{A} \cdot \mathbf{s}_\mathsf{A} + \mathbf{e}_\mathsf{A}$, where $\mathbf{s}_\mathsf{A}, \mathbf{e}_\mathsf{A} \leftarrow \mathcal{D}_\mathsf{A}^\ell \times \mathcal{D}_A^k$, and the sender B samples its public key as $\mathbf{b}_\mathsf{B} := \mathbf{A}^\top \cdot \mathbf{s}_\mathsf{B} + \mathbf{e}_\mathsf{B}$, where $\mathbf{s}_\mathsf{B}, \mathbf{e}_\mathsf{B} \leftarrow \mathcal{D}_\mathsf{B}^k \times \mathcal{D}_B^\ell$.

Again realizing that the values $\mathbf{s}_\mathsf{B}^\top \cdot \mathbf{b}_\mathsf{A}$ and $\mathbf{b}_\mathsf{B}^\top \cdot \mathbf{s}_\mathsf{A} \in \mathcal{R}_q$ are at a short distance, characterized by the MLWE errors, we can extract shared bits for A and B. Thanks to the – by now standard – reconciliation tools in lattice cryptography described in Section 2.6, extraction from the two above values is performed at the cost of additionally exchanging a small ciphertext. In order to have a secure scheme, we also add some masking terms during encapsulation and decapsulation, and extract shared bits from only ctbits coordinates of the above values to limit the leakage induced by ct.

Concretely, in order for B to encapsulate to a counterpart's public key $\mathbf{b}_\mathsf{A}$, B first samples a masking term $y \leftarrow \mathcal{D}_y$ and computes $v := \mathbf{s}_\mathsf{B}^\top \cdot \mathbf{b}_\mathsf{A} + y$. Next, it computes $\mathsf{ct} := \mathsf{HelpRec}_B(v_{1,\ldots,\mathsf{ctbits}})$. The extracted shared key is $K := \mathsf{Rec}_B(v_{1,\ldots,\mathsf{ctbits}}, \mathsf{ct})$, where B is a parameter of the scheme. Given $\mathbf{b}_\mathsf{B}$ and ct, the receiver A is then able to recover the shared secret by computing $v' := \mathbf{b}_\mathsf{B}^\top \cdot \mathbf{s}_\mathsf{A} + y'$, where $y' \leftarrow \mathcal{D}_y$, and recovers $K' := \mathsf{Rec}_B(v'_{1,\ldots,\mathsf{ctbits}}, \mathsf{ct})$.

One can observe that the use of MLWE samples as keys is reminiscent of the standardized KEM Kyber [30], however Sparrow-KEM uses one's private key during encapsulation instead of a fresh random vector so that the shared secret is derived using the sender's key. Sparrow-KEM also differs by using the reconciliation mechanism of FrodoKEM [25] instead of sending a full polynomial as ciphertext. This limits the leakage in case an adversary would act as a man-in-the-middle and send an invalid public key to extract the secret key of the sender, see the Decaps-OW-CPA security analysis in Theorem 4.

3.1 Our construction

We list parameters of Sparrow-KEM in Table 1, and formalize the above construction in Fig. 7.

Parameter	Explanation
$\mathcal{R}_q$	Polynomial ring $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n+1)$
(k, ℓ)	Dimension of the public matrix $\mathbf{A} \in \mathcal{R}_q^{k \times \ell}$
$(\mathcal{D}_\mathsf{A}, \sigma_\mathsf{A})$	Gaussian distribution with width σ_A used for the receiver key sk_A
$(\mathcal{D}_\mathsf{B}, \sigma_\mathsf{B})$	Gaussian distribution with width σ_B used for the sender key sk_B
$(\mathcal{D}_y, \sigma_y)$	Gaussian distribution of width $\sigma_\mathbf{y}$ used for the encapsulation noise $y \in \mathcal{R}_q$
B	Number of shared bits to extract per coordinate
ctbits	Number of coordinates to recover during decapsulation

Table 1: Overview of parameters used in Sparrow-KEM.

Algorithm 1 Sparrow-KEM.Setup()

▷▷▷ Sample a shared public matrix $\mathbf{A} \in \mathcal{R}_q^{k \times \ell}$ ◁◁◁

1: $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{k \times \ell}$
2: **return** $\mathbf{A}$

Figure 6: Setup algorithm of Sparrow-KEM.

Sparrow-KEM differs in two main aspects from the previous works from [7, 8]:

- It leverages MLWE, instead of RLWE or LWE. MLWE allows the design of a very compact scheme, with more flexible choice of parameters than RLWE as n is typically a power of two for practical

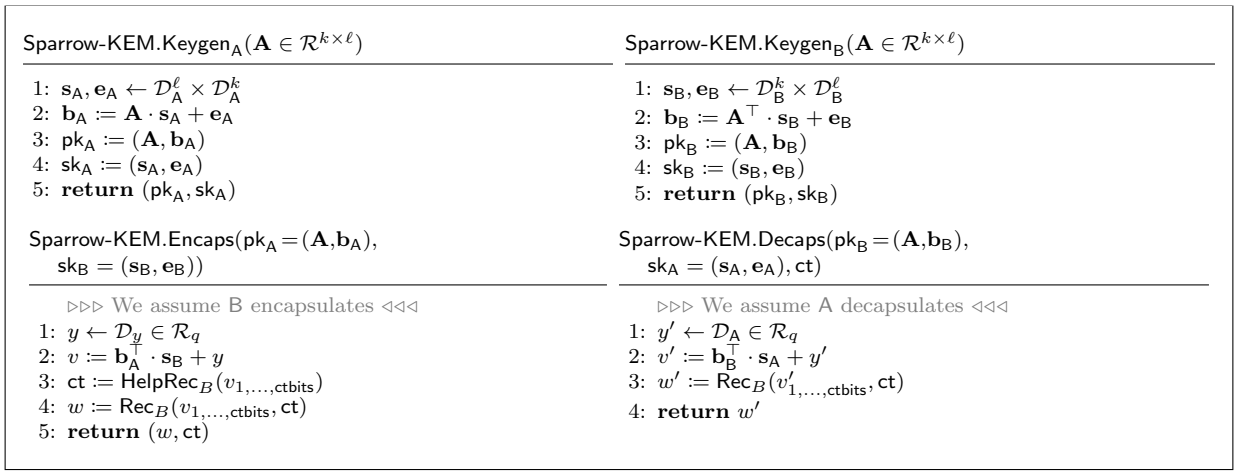

Figure 7: Algorithms of Sparrow-KEM.

implementation. Additionally, leveraging the recent Hint-MLWE problem and its reduction to MLWE, we prove the deniability of our sKEM, which was previously only achieved by relying on LWE in [8].

- The introduction of a masking term in the decapsulation, and the truncation of the ciphertext to ctbits bits allows us to have a provable Decaps-OW-CPA security in the module setting. In particular, truncating the ciphertext to ctbits bits allows us to prove Decaps-OW-CPA with an advantage roughly $2^{\mathsf{ctbits}} \cdot \mathsf{Adv}^{\mathsf{MLWE}}$, while a naive adaptation of K-Waay's proof for Decaps-OW-CPA [8, Lemma 11] would have led to an advantage at least $2^n \cdot \mathsf{Adv}^{\mathsf{MLWE}}$, and an exponential loss that appears hard to compensate with the MLWE security. When ctbits is significantly smaller than n, this leads to a large improvement in parameters. The truncation is notably critical to prove the Decaps-OW-CPA security of Sym-Sparrow-KEM as in that case n is at least a few hundreds.

It is interestingly very easy to define a symmetric version Sym-Sparrow-KEM by relying on RLWE (i.e. $k = \ell = 1$), and taking $\mathcal{D}_A = \mathcal{D}_B$. This comes at the cost of a more constrained parameterization as for practical efficiency we pick $n = 2^k$, but is counterbalanced by a more versatile scheme. Notably, in the context of K-Waay, this allows using a pre-key bundle either for sending or receiving, leading to improved deniability properties. We also believe Sym-Sparrow-KEM could potentially be used in more contexts where Diffie-Hellman was previously used.

3.2 Integrating Sparrow-KEM in K-Waay

K-Waay requires advanced security properties for use in their protocol:

- UNF-1KCA, which ensures that if Alice receives a message allegedly sent by Bob, either Bob really sent it or the decapsulation will fail. This enables implicit authentication.
- IND-1BatchCCA, which is reminiscent of the more usual IND-CCA property. IND-1BatchCCA allows only decapsulation query, but with several keys and ciphertexts at once, and returns either the decapsulated shared keys, or $\perp$ if at least one decapsulation fails.

In the following section, we will only prove that Sparrow-KEM is OW-CPA and Decaps-OW-CPA secure, and rely on the generic split-KEM transform recalled in Fig. 4 and proven in Theorem 1 to obtain a split KEM that is UNF-1KCA and IND-1BatchCCA secure.

This transform is similar to the one introduced in [17], applied to split KEMs. It works at a high-level by appending a hash of the shared key to the ciphertext, ensuring that if the adversary does not know the shared key before encapsulation, decapsulation will fail with overwhelming probability.

Our finalized Sparrow-KEM and Sym-Sparrow-KEM apply the transform with H and H' modeled as random oracles with output size of 2κ bits. We then consider that proving the OW-CPA and Decaps-OW-CPA security of Sparrow-KEM suffices. Notably, we consider that the security loss $Q_H + Q_{H'}$ in Theorem 1 is a proof artifact in [8] due to the use of the random oracle model (ROM) for recovering a secret value and ignore it for practical parameters.

4 Security analysis

This section rigorously assesses the security properties of Sparrow-KEM. We provide formal security proofs of the security notions OW-CPA and Decaps-OW-CPA required for integration in K-Waay. Additionally, we prove the deniability of Sparrow-KEM, even under leakage of the receiver's secret key.

The security analyses presented in this section rely on the standard technique of the Random Oracle Model (ROM). In this idealized model, any cryptographic hash functions used are treated as perfect random oracles, providing a unique, uniformly random output for each distinct input query. While proofs in the ROM do not guarantee security when instantiated with real-world hash functions (which have inherent structures), they provide strong heuristic evidence of security against generic adversaries and significantly simplify the analysis.

4.1 OW-CPA security

Theorem 3. Sparrow-KEM *is* OW-CPA *secure, as defined in Definition 7.*

Formally, there exist adversaries $\mathcal{B}_1$ against the security of the problem $\mathsf{MLWE}_{k,\ell,q,\sigma_\mathsf{A}}$, and $\mathcal{B}_2$ against $\mathsf{MLWE}_{k'=\ell+1,\ell'=k,q,\sigma_\mathsf{B}}$, with running times $\mathcal{T}_{\mathcal{B}_1} \approx \mathcal{T}_{\mathcal{B}_2} \approx \mathcal{T}_{\mathcal{A}}$, such that

$$\mathsf{Adv}^{\mathsf{ow-cpa}}_{\mathsf{Sparrow\text{-}KEM}}(\mathcal{A}) \leqslant \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_1} + \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_2} + 8\varepsilon + 2^{-\mathsf{ctbits}\cdot B}$$

where $\varepsilon \leqslant 1/2$ is such that $\frac{1}{\sigma_\mathsf{B}^2} + \frac{1}{\sigma_y^2 - \sigma_\mathsf{B}^2} \leqslant \eta'_\varepsilon(\mathbb{Z}^n)^{-2}$.

Proof. We proceed with a series of hybrids starting from the OW-CPA game.

Hybrid_1. This is the real OW-CPA game.

Hybrid_2. In this hybrid, we replace the public key pk_A by a random value in $\mathcal{R}$. This reduces to the MLWE problem with standard deviation σ_A.

$$|\mathsf{Adv}^{\mathsf{Hybrid}_1}_{\mathcal{A}} - \mathsf{Adv}^{\mathsf{Hybrid}_2}_{\mathcal{A}}| \leqslant \mathsf{Adv}^{\mathsf{MLWE}_{k,\ell,q,\sigma_\mathsf{A}}}$$

Hybrid_3. In this hybrid, we rewrite the value v in Encaps as:

$$v = (\mathbf{b}_\mathsf{A}^\top \cdot \mathbf{s}_\mathsf{B} + y_1) + y_2$$

where y_1, y_2 are sampled respectively from D_{σ_B} and $D_{\sqrt{\sigma_y^2 - \sigma_\mathsf{B}^2}}$.

The only difference is the separation of y into two Gaussians $y_1 + y_2$. We can apply the convolution lemma from Lemma 2 to show that the statistical distance between the two hybrids is negligible.

Assuming $\frac{1}{\sigma_\mathsf{B}^2} + \frac{1}{\sigma_y^2 - \sigma_\mathsf{B}^2} \leqslant \eta'_\varepsilon(\mathbb{Z}^n)^{-2}$, we obtain by applying Lemma 2:

$$|\mathsf{Adv}^{\mathsf{Hybrid}_2}_{\mathcal{A}} - \mathsf{Adv}^{\mathsf{Hybrid}_3}_{\mathcal{A}}| \leqslant 8\varepsilon$$

Hybrid_4. In this hybrid, we see $\begin{bmatrix} \mathbf{b}_\mathsf{B} \\ \mathbf{b}_\mathsf{A}^\top \cdot \mathbf{s}_\mathsf{B} + y_1 \end{bmatrix} = \begin{bmatrix} \mathbf{A}^\top \\ \mathbf{b}_\mathsf{A}^\top \end{bmatrix} \cdot \mathbf{s}_\mathsf{B} + \begin{bmatrix} \mathbf{e}_\mathsf{B} \\ y_1 \end{bmatrix}$ as an MLWE sample that we replace by a random value in $\mathcal{R}^{\ell+1}$.

This reduces to $\mathsf{MLWE}_{k'=\ell+1,\ell'=k,q,\sigma_\mathsf{B}}$:

$$|\mathsf{Adv}^{\mathsf{Hybrid}_3}_{\mathcal{A}} - \mathsf{Adv}^{\mathsf{Hybrid}_4}_{\mathcal{A}}| \leqslant \mathsf{Adv}^{\mathsf{MLWE}_{k'=\ell+1,\ell'=k,q,\sigma_\mathsf{B}}}$$

Conclusion. At this stage, it is easy to see that the polynomial v in Encaps is uniform in $\mathcal{R}$. According to [8, Lemma 1], the vector w, conditioned on ct, looks uniformly random for the adversary. Hence, its guessing probability is $2^{-\mathsf{ctbits}\cdot B}$.

□

4.2 Decaps-OW-CPA security

We now prove the Decaps-OW-CPA security of Sparrow-KEM. Similarly to the original work [8] introducing Decaps-OW-CPA security, an exponential loss 2^{ctbits} in factor of $\mathsf{Adv}^{\mathsf{MLWE}}$ appears in our proof. In concrete parameters, we can compensate for this loss by targeting a higher MLWE security, and keep negligible the advantage in the proof below. That is, we could for instance asymptotically select $\mathsf{ctbits} = \kappa$, and MLWE parameters so that $\mathsf{Adv}^{\mathsf{MLWE}} \leqslant 2^{-2\kappa}$. Then, $2^{\mathsf{ctbits}} \cdot \mathsf{Adv}^{\mathsf{MLWE}}$ is still a negligible function of the security parameter.

Remark 1. As explained in Section 3.2, the introduction of ctbits improves over a naive loss 2^n that would translate from adapting the proof of [8], and opens the path for a symmetric sKEM that is Decaps-OW-CPA secure. Indeed, n in that case would be at least a few hundreds, while we can select a much lower ctbits at the cost of a slightly increased modulus q.

Theorem 4. Sparrow-KEM *is* Decaps-OW-CPA *secure, as defined in Definition 7.*

Formally, there exist adversaries $\mathcal{B}_1$ against the security of $\mathsf{MLWE}_{k'=\ell,\ell'=k,q,\sigma_\mathsf{B}}$, and $\mathcal{B}_2$ against the security of $\mathsf{MLWE}_{k+1,\ell,q,\sigma_\mathsf{A}}$, and running times $\mathcal{T}_\mathcal{A} \approx \mathcal{T}_{\mathcal{B}_1} \approx \mathcal{T}_{\mathcal{B}_2}$, such that

$$\mathsf{Adv}^{\mathsf{decaps-ow-cpa}}_{\text{Sparrow-KEM}}(\mathcal{A}) \leqslant 2^{\mathsf{ctbits}} \cdot (\mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_1} + \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_2} + \delta^{\mathsf{ctbits}}_{\mathsf{CPA}})$$

where

$$\delta_{\mathsf{CPA}} \coloneqq \max_{\substack{\mathsf{ct}\in\{0,1\} \\ u\in\mathbb{Z}_2^B}} \Pr_{w\xleftarrow{\$}\mathbb{Z}_q} [\mathsf{Rec}_B(2w,\mathsf{ct}) = u]$$

Proof. We proceed with a series of hybrids.

Hybrid_1**.** This is the original Decaps-OW-CPA game.

Hybrid_2**.** In this hybrid, we remove the encapsulation performed by the challenger Section 2.4, and instead sample a random ciphertext ct in $\{0,1\}^{\mathsf{ctbits}}$.

In Hybrid_2, the ciphertext that was computed in Hybrid_1 is sampled with probability $2^{-\mathsf{ctbits}}$, and we deduce that the advantage of the adversary verifies:

$$\mathsf{Adv}^{\mathsf{Hybrid}_1}_{\mathcal{A}} \leqslant 2^{\mathsf{ctbits}}\mathsf{Adv}^{\mathsf{Hybrid}_2}_{\mathcal{A}}$$

Hybrid_3**.** In this hybrid, we replace the public vector $\mathbf{b}_\mathsf{B}$ of Bob by a random value in $\mathcal{R}$.

This reduces to the security of $\mathsf{MLWE}_{k'=\ell,\ell'=k,q,\sigma_\mathsf{B}}$:

$$|\mathsf{Adv}^{\mathsf{Hybrid}_3}_{\mathcal{A}} - \mathsf{Adv}^{\mathsf{Hybrid}_2}_{\mathcal{A}}| \leqslant \mathsf{Adv}^{\mathsf{MLWE}}$$

Hybrid_4**.** In this hybrid, we replace $\begin{bmatrix}\mathbf{b}_\mathsf{A} \\ v'\end{bmatrix}$ by a uniform value in $\mathcal{R}^{k+1}$.

Seeing y' as a MLWE noise of standard deviation σ_A, this reduces to the security of $\mathsf{MLWE}_{k+1,\ell,q,\sigma_\mathsf{A}}$:

$$|\mathsf{Adv}^{\mathsf{Hybrid}_4}_{\mathcal{A}} - \mathsf{Adv}^{\mathsf{Hybrid}_3}_{\mathcal{A}}| \leqslant \mathsf{Adv}^{\mathsf{MLWE}}$$

Conclusion. As v' is now uniform, the probability that any adversary wins in Hybrid_4 is upper-bounded by $\delta^{\mathsf{ctbits}}_{\mathsf{CPA}}$.

The theorem statement follows by combining the hybrids.

□

4.3 Deniability under leakage of receiver's secret key

We study the deniability of Sparrow-KEM under the notion studied in [8], i.e. under leakage of the receiver's secret key.

Theorem 5. Sparrow-KEM *is deniable, with the encapsulation simulator from Fig. 8.*

Formally, there exists an adversary $\mathcal{B}$ against the security of $\mathsf{Hint\text{-}MLWE}_{k'=\ell,\ell'=k,q,Q=1,\sigma_\mathsf{B},\sigma_y,\mathcal{C}=D_{\sigma_\mathsf{A}}}$ *(i.e. with a hint on Bob's secret of the form $h = \mathbf{e}_\mathsf{A}^\top \cdot \mathbf{s} - \mathbf{s}_\mathsf{A}^\top \cdot \mathbf{e} + y$), and running time $\mathcal{T}_\mathcal{B} \approx \mathcal{T}_\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathsf{deny}}_{\mathsf{Sparrow\text{-}KEM},\mathsf{SimEncaps}}(\mathcal{A}) \leqslant \mathsf{Adv}^{\mathsf{Hint\text{-}MLWE}}_{\mathcal{B}}$$

Remark 2. Our security model assumes that keys are honestly generated to provide deniability. If the receiver A generates its key adversarially, she may able to prove that a message was sent by Bob.

$\mathsf{SimEncaps}(\mathsf{sk}_\mathsf{A} \coloneqq (\mathbf{s}_\mathsf{A}, \mathbf{e}_\mathsf{A}), \mathsf{pk}_\mathsf{B} \coloneqq (\mathbf{A}, \hat{\mathbf{b}}_\mathsf{B}))$

1: $\mathbf{s}_\mathsf{sim}, \mathbf{e}_\mathsf{sim} \leftarrow \mathcal{D}_\mathsf{B}^k \times \mathcal{D}_\mathsf{B}^\ell$
2: $y_\mathsf{sim} \leftarrow \mathcal{D}_y$
3: $v_\mathsf{sim} \coloneqq \mathbf{s}_\mathsf{sim}^\top \cdot \mathbf{e}_\mathsf{A} - \mathbf{e}_\mathsf{sim}^\top \cdot \mathbf{s}_\mathsf{A} + \mathbf{b}_\mathsf{B}^\top \cdot \mathbf{s}_\mathsf{A} + y_\mathsf{sim}$
4: $\mathsf{ct} \coloneqq \mathsf{HelpRec}_B(v_\mathsf{sim})$
5: $w \coloneqq \mathsf{Rec}_B(w, \mathsf{ct})$
6: **return** (w, ct)

Figure 8: Deniability simulator

Proof. We proceed with a series of hybrids.

Hybrid_1**.** This is the original deniability game $\mathsf{DENY}^{\mathsf{REAL}}$, where the message is encapsulated with the private key of Bob.

Hybrid_2**.** In this second hybrid, we represent the computation of v as $v \coloneqq \mathbf{s}_\mathsf{B}^\top \cdot \mathbf{e}_\mathsf{A} - \mathbf{e}_\mathsf{B}^\top \cdot \mathbf{s}_\mathsf{A} + \mathbf{b}_\mathsf{B}^\top \cdot \mathbf{s}_\mathsf{A} + y$, indeed:

$$\begin{aligned} v &= \mathbf{s}_\mathsf{B}^\top \cdot \mathbf{b}_\mathsf{A} + y \\ &= \mathbf{s}_\mathsf{B}^\top \cdot (\mathbf{A} \cdot \mathbf{s}_\mathsf{A} + \mathbf{e}_\mathsf{A}) + y \\ &= \mathbf{s}_\mathsf{B}^\top \cdot \mathbf{A} \cdot \mathbf{s}_\mathsf{A} + \mathbf{s}_\mathsf{B}^\top \cdot (\mathbf{e}_\mathsf{A}) + y \\ &= (\mathbf{b}_\mathsf{B}^\top - \mathbf{e}_\mathsf{B}^\top) \cdot \mathbf{s}_\mathsf{A} + \mathbf{s}_\mathsf{B}^\top \cdot \mathbf{e}_\mathsf{A} + y \end{aligned}$$

There is no advantage difference,

$$\mathsf{Adv}^{\mathsf{Hybrid}_1}_{\mathcal{A}} = \mathsf{Adv}^{\mathsf{Hybrid}_2}_{\mathcal{A}}$$

Hybrid_3**.** In this hybrid, we replace $\mathbf{b}_\mathsf{B}$ by a random vector, independent from $\mathbf{s}_\mathsf{B}, \mathbf{e}_\mathsf{B}$. This reduces to the Hint-MLWE problem by observing that we can simulate the adversary's view with only the public key $\mathbf{b}_\mathsf{B}$ and a hint on Bob's secret $h = \mathbf{s}_\mathsf{B}^\top \cdot \mathbf{e}_\mathsf{A} - \mathbf{e}_\mathsf{B}^\top \cdot \mathbf{s}_\mathsf{A} + y$.

Thus, there exists an adversary $\mathcal{B}_1$ against the security of the problem $\mathsf{Hint\text{-}MLWE}_{k'=\ell,\ell'=k,q,Q=1,\sigma_\mathsf{B},\sigma_y,\mathcal{C}=D_{\sigma_\mathsf{A}}}$, i.e. with a hint on Bob's secret of the form $h = \mathbf{s}_\mathsf{B}^\top \cdot \mathbf{e}_\mathsf{A} - \mathbf{e}_\mathsf{B}^\top \cdot \mathbf{s}_\mathsf{A} + y$, such that

$$|\mathsf{Adv}^{\mathsf{Hybrid}_2}_{\mathcal{A}} - \mathsf{Adv}^{\mathsf{Hybrid}_3}_{\mathcal{A}}| \leqslant \mathsf{Adv}^{\mathsf{Hint\text{-}MLWE}}_{\mathcal{B}_1}$$

Conclusion. We notice that the last hybrid corresponds to the behavior of our SimEncaps from Fig. 8, and to the game $\mathsf{DENY}^{\mathsf{SIM}}$. It concludes our proof.

□

5 Concrete parameters

In this section, we give concrete parameters for our two schemes Sparrow-KEM and Sym-Sparrow-KEM. We aim for an overwhelming correctness, i.e. decapsulation always succeeds for an honest transcript, as well as simultaneously achieving OW-CPA, Decaps-OW-CPA security, and deniability. Applying the generic transform from Section 3.2, we obtain split KEMs achieving UNF-1KCA and IND-1BatchCCA security and can be plugged in K-Waay.

5.1 Correctness

In order to achieve an overwhelming correctness, we need to ensure that the values v_{A} and v_{B} computed by Alice and Bob are at a close distance. Precisely, when q is odd, Lemma 1 states that Alice and Bob recovers the same shared secret in $\{0,1\}^{\mathsf{ctbits}\cdot B}$ if:

$$\|v_{\mathsf{A}} - v_{\mathsf{B}}\|_\infty \leqslant \left\lfloor \frac{q}{2^{B+2}} \right\rfloor$$

where B is the number of bits recovered from each coordinate.

When both parties behave correctly:

- $v_{\mathsf{B}} \coloneqq \mathbf{s}_{\mathsf{B}}^\top \cdot \mathbf{b}_{\mathsf{A}} + y = \mathbf{s}_{\mathsf{B}}^\top \cdot (\mathbf{A} \cdot \mathbf{s}_{\mathsf{A}} + \mathbf{e}_{\mathsf{A}}) + y = \mathbf{s}_{\mathsf{B}}^\top \cdot \mathbf{A} \cdot \mathbf{s}_{\mathsf{A}} + \underbrace{(\mathbf{s}_{\mathsf{B}}^\top \cdot \mathbf{e}_{\mathsf{A}} + y)}_{\text{small}}$
- $v_{\mathsf{A}} \coloneqq \mathbf{b}_{\mathsf{B}}^\top \cdot \mathbf{s}_{\mathsf{A}} + y' = (\mathbf{s}_b^\top \cdot \mathbf{A} + \mathbf{e}_{\mathsf{B}}^\top) \cdot \mathbf{s}_{\mathsf{A}} + y' = \mathbf{s}_{\mathsf{B}}^\top \cdot \mathbf{A} \cdot \mathbf{s}_{\mathsf{A}} + \underbrace{(\mathbf{e}_{\mathsf{B}}^\top \cdot \mathbf{s}_{\mathsf{A}} + y')}_{\text{small}}$

We have

$$v_{\mathsf{A}} - v_{\mathsf{B}} = \mathbf{e}_{\mathsf{B}}^\top \cdot \mathbf{s}_{\mathsf{A}} + y' - (\mathbf{s}_{\mathsf{B}}^\top \cdot \mathbf{e}_{\mathsf{A}} + y)$$

For concrete parameters, we adapted https://github.com/pq-crystals/security-estimates to exactly evaluate the distribution of the coordinates of $v_{\mathsf{A}} - v_{\mathsf{B}}$, and find an overwhelming bound β (probability $\geqslant 1 - 2^{-\kappa}$) on $\|v_{\mathsf{A}} - v_{\mathsf{B}}\|_\infty$. We then take for B the maximal value such that $\beta \leqslant \left\lfloor \frac{q}{2^{B+2}} \right\rfloor$, ensuring correct decapsulation with overwhelming probability.

We make our script available at https://github.com/GuilhemN/sparrow-kem/blob/main/security/correctness/eval_sparrow.py.

5.2 Security evaluation

The deniability, OW-CPA and Decaps-OW-CPA properties of Sparrow-KEM and Sym-Sparrow-KEM all reduce to MLWE and Hint-MLWE.

In more detail, our proofs state that Sparrow-KEM and Sym-Sparrow-KEM are secure if the shared secret has sufficient entropy, and if the following problems are hard:

- For OW-CPA, $\mathsf{MLWE}_{k,\ell,q,\sigma_{\mathsf{A}}}$, and $\mathsf{MLWE}_{k'=\ell+1,\ell'=k,q,\sigma_{\mathsf{B}}}$.
- For deniability, $\mathsf{Hint\text{-}MLWE}_{k'=\ell,\ell'=k,q,Q=1,\sigma_{\mathsf{B}},\sigma_y,\mathcal{C}=D_{\sigma_{\mathsf{A}}}}$ (i.e. with a hint $h = \mathbf{e}_{\mathsf{A}}^\top \cdot \mathbf{s} - \mathbf{s}_{\mathsf{A}}^\top \cdot \mathbf{e} + y$, for $\mathbf{s} \leftarrow D_{\sigma_{\mathsf{B}}}, \mathbf{y} \leftarrow D_{\sigma_y}$).

- For Decaps-OW-CPA, we need $\mathsf{MLWE}_{k'=\ell,\ell'=k,q,\sigma_\mathsf{B}}$ and $\mathsf{MLWE}_{k+1,\ell,q,\sigma_\mathsf{A}}$. However, our proof of Theorem 4 loses a factor 2^{ctbits} in front of the MLWE advantage and shared secret entropy. It can either be compensated by a higher MLWE security, and larger shared secret entropy, or we can instead establish an heuristic analysis of the concrete Decaps-OW-CPA security. We provide both proven and heuristic parameter sets below.

We are ultimately interested in the properties UNF-1KCA and IND-1BatchCCA recalled in Section 2.3 to use our split KEMs directly in K-Waay. As stated previously, we rely on the transform from Section 3.2. Theorem 1 proves that the UNF-1KCA and IND-1BatchCCA security of this transform derive from the OW-CPA and Decaps-OW-CPA security of the initial split KEM. After taking hash outputs of bitsize 2κ, the transform only loses some security due to the use of the ROM for recovering a secret value with guessing arguments; as typically done when selecting concrete parameters, we consider that the loss induced by these guessing arguments are proof artifacts and focus on the security of the OW-CPA and Decaps-OW-CPA properties for Sparrow-KEM and Sym-Sparrow-KEM.

Target security. We select parameters for our scheme aiming for NIST security level I, i.e. breaking the security of our scheme should be as hard as breaking AES-128. Concretely, we consider as metric the bit-security — the maximal advantage of an adversary multiplied by its execution time ($\max_{\mathcal{A}} \log_2(\mathsf{Adv}(\mathcal{A}) \cdot \mathcal{T}_{\mathcal{A}})$), and make sure it is higher than for AES.

Concrete security of MLWE and Hint-MLWE. In order to evaluate the concrete hardness of MLWE and Hint-MLWE, and hence the security of our schemes, we make use of the *lattice-estimator*, available at https://github.com/malb/lattice-estimator. It estimates the security of MLWE parameter sets against the best known attacks. We provide security estimates for our selected parameters following the "Core-SVP" methodology from [1].

In particular, for the Hint-MLWE assumption that underlies our sKEM deniability, we leverage Theorem 2 which states that breaking $\mathsf{Hint\text{-}MLWE}_{k'=\ell,\ell'=k,q,Q=1,\sigma_\mathsf{B},\sigma_y,\mathcal{C}=D_{\sigma_\mathsf{A}}}$ is as hard as breaking $\mathsf{MLWE}_{k'=\ell,\ell'=k,q,\sigma_\mathsf{red}}$ where

$$\sigma_\mathsf{red}^2 = \frac{1}{2} \cdot \left(\frac{1}{\sigma_\mathsf{B}^2} + \frac{s_1\left(\begin{bmatrix}\mathbf{e}_\mathsf{A}\\ \mathbf{s}_\mathsf{A}\end{bmatrix}\right)^2}{\sigma_y^2} \right)^{-1}$$

Additionally, we can precisely evaluate $s_1\left(\begin{bmatrix}\mathbf{e}_\mathsf{A}\\ \mathbf{s}_\mathsf{A}\end{bmatrix}\right)$ using Lemma 3.

Smoothing parameter in Theorem 3. The smoothing parameter condition in Theorem 3 is a proof artifact introduced to rely on the hardness of MLWE where the error has the same standard deviation as the secret. This is merely done for clarity, and we can safely ignore it for concrete parameters considering that we could have instead relied on a harder instance of MLWE with an error of larger standard deviation.

5.2.1 Parameter Sets following Theorem 4

We start by selecting parameters such that the bit-security indicated by Theorems 3, 4, and 5 correspond to NIST level I.

In particular, we take into account the exponential advantage loss 2^{ctbits} in Theorem 4 by having a larger message entropy, and higher MLWE security.

We include these proven parameter sets in Table 2. These parameters are already quite practical with a public key size of 2.5kB for an asymmetric version of Sparrow-KEM, and 2.8kB for a symmetric version.

5.2.2 Heuristic Parameter Sets

Although we could use proven parameters from the previous section, the exponential loss in Theorem 4 degrades parameters while it does not lead to a concrete attack. We provide in this section a tighter

Scheme	$\lvert\mathsf{pk}_\mathsf{A}\rvert$	$\lvert\mathsf{pk}_\mathsf{B}\rvert$	$\lvert\mathsf{ct}\rvert$	Security C/Q	Deniability C/Q
Sparrow-KEM					
Proved	2 592B	2 592B	40B	130 /102	147/127
Heuristic	2 016B	2 016B	48B	135/115	123/106
Sym-Sparrow-KEM					
Proved	2 816B	2 816B	36B	112/91	91/78
Heuristic	2 176B	2 176B	48B	154/132	113/97
FrodoKex+ [8]	21 300B	21 300B	136B		

Table 2: Parameter sets for Sparrow-KEM based on the security reduction ("*Proved*") and based on our concrete cryptanalysis ("*Heuristic*"), aiming for NIST security level I. We indicate the core-SVP hardness [1] for the deniability, OW-CPA, and Decaps-OW-CPA security properties (hardness classical/quantum), minus the dimensions for free [10]. The core-SVP methodology ignores several polynomial factors, and is often used in the literature to compare lattice-based constructions. As a comparison, Dilithium-II [22] achieves 112/98 bits of core-SVP hardness. We provide detailed parameter sets in Tables 5 and 6. As a comparison to prior state-of-the-art, we indicate sizes for FrodoKex+ [8].

heuristic analysis of the Decaps-OW-CPA security of Sparrow-KEM to remove the factor 2^{ctbits} in front of the MLWE advantage.

Concrete security of Decaps-OW-CPA. We recall that the ctbits bits loss in the Decaps-OW-CPA proof comes from the replacement of the encapsulated ciphertext by a random value in $\{0,1\}^{\mathsf{ctbits}}$. This shortcuts what information is actually provided to the adversary by ct, by simply simulating it.

Instead, let's look in more detail what could leak to the adversary through the observation of ct. ct is obtained as the result of encapsulating to a malicious public key $\mathsf{pk}_\mathcal{A}$. The adversary then learns n bits of $\mathsf{pk}_\mathcal{A}^\top \cdot \mathbf{s}_\mathsf{B} + y$.

This leak can heuristically be exploited in two ways:

1. Learn information about the shared secret value $v \coloneqq \mathbf{s}_\mathsf{A}^\top \mathbf{A}^\top \mathbf{s}_\mathsf{B}$.
2. Learn enough information about $\mathbf{s}_\mathsf{B}$ and $b_\mathsf{B} \coloneqq \mathbf{A}^\top \cdot \mathbf{s}_\mathsf{B} + \mathbf{e}_\mathsf{B}$ so that $\mathbf{b}_\mathsf{B}^\top \mathbf{s}_\mathsf{A} + y'$ is no longer pseudo-randomly uniformly distributed.

For (1), we cover this attack by counting a loss of ctbits bits in the entropy of the shared secret. A slight increase of the bound B suffices to counter it.

For (2), observe that the knowledge of $\mathsf{pk}_\mathcal{A}^\top \cdot \mathbf{s}_\mathsf{B}$ leaks the value of $\mathbf{s}_\mathsf{B}$ in at most ctbits directions. We thus consider that no information leaks on the remaining $\ell \cdot n - \mathsf{ctbits}$ dimensions of $\mathbf{s}$. In the selective case, where $\mathsf{pk}_\mathcal{A}$ is chosen before $\mathbf{A}$ and pk_B are provided to the adversary, since $\mathbf{s}_\mathsf{B}$ is sampled for a spherical Gaussian, we consider that the hardness of distinguishing $\mathbf{A}^\top \cdot \mathbf{s}_\mathsf{B} + \mathbf{e}_\mathsf{B}$ from uniform can be evaluated from $\mathsf{LWE}_{\ell \cdot n - \mathsf{ctbits}, k \cdot n, \sigma_\mathsf{B}}$ (i.e. loss of ctbits dimensions of the secret). For the case where $\mathsf{pk}_\mathcal{A}$ is adaptatively chosen, we consider that analogously at least $n \cdot (\ell - 1)$ dimensions of $\mathbf{b}_\mathsf{B}$ are pseudo-random and suffices to have $\mathbf{b}_\mathsf{B}^\top \cdot \mathbf{s}_\mathsf{A} + y'$ uniformly distributed.

Following the above arguments, the heuristic advantage of an adversary $\mathcal{A}$ against the Decaps-OW-CPA security of Sparrow-KEM is:

$$\mathsf{Adv}^{\mathsf{LWE}}_{\mathcal{B}_1} + \mathsf{Adv}^{\mathsf{LWE}}_{\mathcal{B}_2} + 2^{\mathsf{ctbits}} \cdot \delta^{\mathsf{ctbits}}_{\mathsf{CPA}}$$

where $\mathcal{B}_1$ is an adversary against the security of $\mathsf{LWE}_{k'=n\cdot\ell,\ell'=n\cdot k-\mathsf{ctbits},\sigma_\mathsf{B}}$, and $\mathcal{B}_2$ is an adversary against the security of $\mathsf{LWE}_{n\cdot(k+1),n\cdot\ell-\mathsf{ctbits},\sigma_\mathsf{A}}$. Also,

$$\delta_{\mathsf{CPA}} \coloneqq \max_{\substack{\mathsf{ct}\in\{0,1\} \\ u\in\mathbb{Z}_2^B}} \Pr_{w\xleftarrow{\$}\mathbb{Z}_q} [\mathsf{Rec}_B(2w, \mathsf{ct}) = u]$$

The above heuristic analysis allows us to determine more compact parameter sets for Sparrow-KEM and for Sym-Sparrow-KEM, by limiting the increase of the MLWE parameters. We provide parameter sets

following our analysis in Table 2. As our heuristic analysis of the Decaps-OW-CPA security remains quite convervative, we recommend using these parameter sets over the ones set fully based on the security proofs.

Note that Sparrow-KEM has a public key size of only 2016 bytes, i.e. 10.5× smaller than the previous state of the art split KEM FrodoKex+.

6 Evaluation of Sparrow-KEM within the full protocol K-Waay

To assess the practical impact of Sparrow-KEM, we evaluate its performance within the K-Waay protocol and compare it to concurrent approaches for a post-quantum Signal key exchange protocol based on ring signatures. We evaluate two key metrics for a post-quantum X3DH: communication overhead, and computational cost. The communication overhead is primarily determined by the size of long term keys, pre-key bundles, and ciphertexts, while the computational cost is typically dominated by the cost of the split KEM or ring signature used.

We will consider three protocols for our evaluation: the ring signature based protocols HKKP [16] and SPQR [6], and the split KEM based protocol K-Waay [8]. We start with an overview of the components used in each of these protocols.

Overview of HKKP. HKKP [16] uses ring signatures in order to build a deniable post-quantum X3DH protocol. They derive session keys from both a long term KEM, and an ephemeral KEM. The public key of the ephemeral KEM is provided as part of the pre-key bundle of the receiver, at the same time as a ring verification key. We consider HKKP with signed pre-keys as is the case in current Signal's X3DH; this ensures stronger forward-secrecy guarantees.

- The long term key is composed of: (i) *a long-term KEM key*, (ii) *a long-term ring signature verification key*, (iii) *a long-term signature verification key*.
- The receiver pre-key bundle is composed of: (i) *an ephemeral KEM public key*, (ii) *an ephemeral ring signature verification key*, (iii) *a signature*.
- The ciphertext is composed of: (i) *a long-term KEM ciphertext*, (ii) *an ephemeral KEM ciphertext*, (iii) *a ring signature*.

Overview of SPQR. SPQR [6] relies on Designated Verifier Signature (DVS) for its post-quantum X3DH protocol. DVS were proven equivalent to ring signatures in the two-parties setting in [6, 16]. Similarly to HKKP, we consider that pre-key bundles are signed for our comparison.

- The long term key is composed of: (i) *a long-term KEM key*, (ii) *a long-term DVS verification key*, (iii) *a long-term signature verification key*.
- The receiver pre-key bundle is composed of: (i) *a semi-static KEM public key*, (ii) *an ephemeral KEM public key*, (iii) *an ephemeral DVS verification key*, (iv) *a signature*.
- The ciphertext is composed of: (i) *a long-term KEM ciphertext*, (ii) *a semi-static KEM ciphertext*, (iii) *an ephemeral KEM ciphertext*, (iv) *a DVS signature*.

Overview of K-Waay. K-Waay [8] is built using a deniable split KEM. The protocol differs from the two previous constructions in that both the sender and receiver upload a pre-key bundle, which is retrieved by the counterpart for the key derivation. K-Waay also relies on a long-term KEM, and an ephemeral KEM to provide both key compromise resistance, and forward secrecy. The split KEM keys are used for implicit authentication of the parties and are included in pre-key bundles.

- The long term key is composed of: (i) *a long-term KEM key*, (ii) *a long-term signature verification key*.

- The receiver pre-key bundle is composed of: (i) *an ephemeral KEM public key*, (ii) *an ephemeral receiver split KEM key*, (iii) *a signature*.
- The sender pre-key bundle is composed of: (i) *an ephemeral sender split KEM key*, (ii) *a signature*.
- The ciphertext is composed of: (i) *a long-term KEM ciphertext*, (ii) *an ephemeral KEM ciphertext*, (iii) *an ephemeral split KEM ciphertext*.

6.1 Communication

We now evaluate the communication overhead of each protocol.

Instantiating individual components. We select components targetting NIST level I. We assume for our evaluation that signatures are instanciated with Falcon-512 [28], and both long-term and ephemeral KEMs are instanciated with Kyber-512 [30].

For the ring signatures, a large number of choices appear, providing different tradeoffs in terms of public key size versus signature size. We consider the lattice-based ring signatures Falafl [4], DualRing-LB [33], Raptor [20] and Gandalf [12]. We additionally consider the isogeny-based ring signature Calamari [4], although the comparison should be made cautiously as isogenies provide different tradeoffs than lattices (in the security assumption, compactness, efficiency). We only provide sizes for SPQR with Raptor and Gandalf for conciseness, as the total size is always higher than when using HKKP with the same ring signature scheme.

For the split KEMs, we compare the Frodo-based split KEM FrodoKex+ from [8], to our Sparrow-KEM and Sym-Sparrow-KEM.

Scheme	$\lvert\mathsf{lpk}\rvert$	$\lvert\mathsf{prek}_\mathsf{A}\rvert$	$\lvert\mathsf{prek}_\mathsf{B}\rvert$	$\lvert\mathsf{ct}\rvert$	$\downarrow$ **total**
HKKP + Gandalf	2 597	2 366	–	2 772	**7 735**
HKKP + Calamari	1 761	1 530	–	5 198	**8 489**
HKKP + Raptor	2 597	2 366	–	4 068	**9 031**
SPQR + Gandalf	2 597	3 166	–	3 540	**9 303**
K-Waay + Sparrow-KEM	1 697	3 514	2 714	1 584	**9 509**
K-Waay + Sym-Sparrow-KEM	1 697	3 674	2 874	1 584	**9 829**
SPQR + Raptor	2 597	3 166	–	4 836	**10 599**
HKKP + DualRing-LB	4 193	3 962	–	5 413	**13 568**
HKKP + Falafl	5 793	5 562	–	31 552	**42 907**
K-Waay + FrodoKex+	1 697	23 246	22 446	1 632	**49 021**

Table 3: Size comparison in bytes between K-Waay, HKKP [16] and SPQR [6], targetting NIST level I. We computed the sizes for both HKKP and SPQR implemented with signed prekey bundles. The total is the sum of all the previous columns, and corresponds to the total communication (upload + download) of a party for performing one key exchange.

We summarize our results in Table 3. The use of Sparrow-KEM within K-Waay shrinks the total communication of the protocol by a factor 5.1× to only 9.5kB, and significantly improves its practicality. This update also makes K-Waay compete with the ring signature based protocols HKKP and SPQR in terms of communication size: K-Waay with Sparrow-KEM exchanges the same order of data as HKKP and SPQR combined with the shortest post-quantum ring signatures available.

6.2 Speed benchmarks

Collins et al. [8] observed that the non-standard primitive (ring signature or split KEM) is the bottleneck in post-quantum X3DH. This can be explained as post-quantum KEMs are typically inexpensive to run, as opposed to the expensive computations used in the shortest ring signatures for trapdoor sampling (lattice-based), cut-and-choose (hash functions) for Falafl, isogenies computations for CSIDH, or the matrix multiplication in FrodoKex+. Note that the cost of the key generation of the ring signature scheme or split KEM is as important as the other algorithms in the case of X3DH. Indeed, these primitives are used for ephemeral values, and thus keys are sampled for every session as part of a prekey bundle and then used only once.

We focus in this section on studying the performance improvement provided by Sparrow-KEM, compared to previous works on split KEMs and ring signatures.

We implemented Sparrow-KEM in C[1]. We implemented Montgomery modular reduction, an NTT for fast ring multiplications, and table-based discrete Gaussian samplers for sampling the secrets and noises. We compare its efficiency to existing implementations of FrodoKex+, and against the ring signatures Raptor, Calamari and Falafl. DualRing-LB and Gandalf do not provide implementations, hence, we measured the performance of Falcon 512 as a baseline, and as a likely upper bound on future performance improvements of short ring signature implementations.

We provide our measurements in Table 4, performed on a Ryzen Pro 7 5850U-based @ 3GHz based laptop. We observe a 40× performance improvement of using Sparrow-KEM over FrodoKex+, which was already previously shown to significantly outperform Raptor. Sparrow-KEM even outperforms Falcon, and is 6× faster. Thus, even assuming significant optimizations in the implementation of the aforementioned ring signature schemes, it is likely that K-Waay with Sparrow-KEM will keep outperforming X3DH based on ring signatures.

Split KEM	KeygenA	KeygenB	Encaps	Decaps	**total (M cycles)**
FrodoKex+	60.8	59.8	0.4	0.3	**121.3**
Sparrow-KEM	0.7	0.7	1.5	0.1	**3.0**

Ring Signature	Keygen	Sign	Verify	**total (M cycles)**
Raptor[2]	53.5	95.5	80.9	**229.9**
Calamari	119.5	46 581	41 250	**87 950.5**
Falafl[3]	0.1	163.2	76.4	**239.7**
Falcon 512	17.2	0.6	0.1	**17.9**

Table 4: Performance comparison between the split KEMs FrodoKex+, Sparrow-KEM, and the ring signatures Raptor, Calamari and Falafl, targetting NIST level I. We also provide measurements for the signature scheme Falcon-512 as a baseline for ring signatures. Measurements are expressed in Mega cycles. The total corresponds to the online computation cost – i.e. the cost of generating and interpreting prekey bundles and ciphertexts – of the aforementioned primitive in K-Waay, HKKP or SPQR. We include the cost of key generation, as the ring signature and split KEM keys are ephemerals sampled for each session, and included in a pre-key bundle.

7 Discussion on advantages and limitations

This section delves into a comparative analysis of using our improved instantiation of split KEM and K-Waay for a post-quantum X3DH protocol, and the approach based on ring signatures. We focus on key performance metrics, and practical considerations to elucidate the advantages and limitations of each approach.

[1] Our code is available at https://github.com/GuilhemN/sparrow-kem.
[2] https://github.com/zhenfeizhang/raptor/tree/37d78152bbca69b27b13a184347ceaec8ffbd69f
[3] https://github.com/WardBeullens/Calamari-and-Falafl/tree/38a94bbfa39c211dbf10abc65b5aec7a0b9bc070

Communication overhead. Sparrow-KEM significantly improves the communication cost of K-Waay to the point that the two approaches to build a post-quantum X3DH based on a split KEM or ring signature appear closeby. The total communication overhead of the proposed protocols differ only by a small margin, depending on the exact underlying primitives used. Communication overhead is thus no longer a significantly differentiating metric in favor of the ring signature approach, and K-Waay is a competitive proposal.

Security model and deniability. While both split-KEMs and ring signatures offer post-quantum security, their use in a post-quantum X3DH differ in their underlying security models and the level of deniability they provide. The K-Waay protocol, relying on split-KEMs, inherently involves two pre-key bundles, which might introduce additional complexities when arguing deniability compared to single-bundle approaches based on ring signatures which only introduce a pre-key bundle for the receiver. Pratical deniability of each of these models however rather seems to be a legal question, as well as dependent on the broader protocol implementation – for instance whether users are authenticated by the server when sending their ciphertext. It is interesting further work to explore practical deniability [29, 32].

Efficiency and flexibility. Split-KEMs generally have simpler and more efficient designs compared to ring signatures. [8] observed that K-Waay with FrodoKex+ was faster than ring signatures based schemes. Sparrow-KEM increases this gap and is 40× faster than FrodoKex+. We also remark that secure messaging applications are likely to be deployed in adversarional environments, where side-channel countermeasures are desirable. Masking techniques countermeasures can be more readily applied to split KEMs compared to ring signatures, providing stronger confidence in the protocol.

Modular integration. While Sparrow-KEM was designed to optimize K-Waay, the Sparrow-KEM themselves are cryptographic primitives. Other protocols could potentially integrate these KEMs as building blocks without needing to adopt the entire K-Waay protocol structure. They could leverage the technique (the efficient, deniable split KEM construction) independently, thus benefiting from the paper's core innovation without direct dependence on K-Waay as a complete protocol framework. We notably could expect variants of the key-exchange protocols based on ring signatures [16, 6], to leverage split KEMs.

8 Acknowledgments

We would like to thank Loïs Huguenin-Dumittan for his help to start this project, Thomas Espitau for insightful discussions on the concrete security analysis of the Decaps-OW-CPA security, and Thomas Prest for his result on the spectral norm of Gaussian samples.

References

[1] Erdem Alkim et al. "Post-quantum Key Exchange - A New Hope". In: *USENIX Security 2016*. Ed. by Thorsten Holz and Stefan Savage. USENIX Association, Aug. 2016, pp. 327–343.

[2] W. Banaszczyk. "New bounds in some transference theorems in the geometry of numbers". In: *Mathematische Annalen* 296.1 (Dec. 1993), pp. 625–635. ISSN: 1432-1807. DOI: 10.1007/BF01445125. URL: https://doi.org/10.1007/BF01445125.

[3] Andrea Basso. *POKE: A Framework for Efficient PKEs, Split KEMs, and OPRFs from Higher-dimensional Isogenies*. Cryptology ePrint Archive, Paper 2024/624. https://eprint.iacr.org/2024/624. 2024. URL: https://eprint.iacr.org/2024/624.

[4] Ward Beullens, Shuichi Katsumata, and Federico Pintore. "Calamari and Falafl: Logarithmic (Linkable) Ring Signatures from Isogenies and Lattices". In: *ASIACRYPT 2020, Part II*. Ed. by Shiho Moriai and Huaxiong Wang. Vol. 12492. LNCS. Springer, Heidelberg, Dec. 2020, pp. 464–492. DOI: 10.1007/978-3-030-64834-3_16.

[5] Joppe W. Bos et al. "Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE". In: *ACM CCS 2016*. Ed. by Edgar R. Weippl et al. ACM Press, Oct. 2016, pp. 1006–1018. DOI: 10.1145/2976749.2978425.

[6] Jacqueline Brendel et al. "Post-quantum Asynchronous Deniable Key Exchange and the Signal Handshake". In: *PKC 2022, Part II*. Ed. by Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe. Vol. 13178. LNCS. Springer, Heidelberg, Mar. 2022, pp. 3–34. DOI: 10.1007/978-3-030-97131-1_1.

[7] Jacqueline Brendel et al. "Towards Post-Quantum Security for Signal's X3DH Handshake". In: *SAC 2020*. Ed. by Orr Dunkelman, Michael J. Jacobson Jr., and Colin O'Flynn. Vol. 12804. LNCS. Springer, Heidelberg, Oct. 2020, pp. 404–430. DOI: 10.1007/978-3-030-81652-0_16.

[8] Daniel Collins et al. *K-Waay: Fast and Deniable Post-Quantum X3DH without Ring Signatures*. Cryptology ePrint Archive, Paper 2024/120. https://eprint.iacr.org/2024/120. 2024. URL: https://eprint.iacr.org/2024/120.

[9] Mario Di Raimondo, Rosario Gennaro, and Hugo Krawczyk. "Deniable authentication and key exchange". In: *ACM CCS 2006*. Ed. by Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati. ACM Press, Oct. 2006, pp. 400–409. DOI: 10.1145/1180405.1180454.

[10] Léo Ducas. "Shortest Vector from Lattice Sieving: A Few Dimensions for Free". In: *EUROCRYPT 2018, Part I*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10820. LNCS. Springer, Heidelberg, Apr. 2018, pp. 125–145. DOI: 10.1007/978-3-319-78381-9_5.

[11] Thomas Espitau, Guilhem Niot, and Thomas Prest. *Flood and Submerse: Distributed Key Generation and Robust Threshold Signature from Lattices*. Cryptology ePrint Archive, Paper 2024/959. https://eprint.iacr.org/2024/959. 2024. URL: https://eprint.iacr.org/2024/959.

[12] Phillip Gajland, Jonas Janneck, and Eike Kiltz. *Ring Signatures for Deniable AKEM: Gandalf's Fellowship*. Cryptology ePrint Archive, Paper 2024/890. https://eprint.iacr.org/2024/890. 2024. URL: https://eprint.iacr.org/2024/890.

[13] Nicholas Genise et al. "Improved Discrete Gaussian and Subgaussian Analysis for Lattice Cryptography". In: *PKC 2020, Part I*. Ed. by Aggelos Kiayias et al. Vol. 12110. LNCS. Springer, Heidelberg, May 2020, pp. 623–651. DOI: 10.1007/978-3-030-45374-9_21.

[14] Brian Goncalves and Atefeh Mashatan. "Towards Generalized Diffie-Hellman-esque Key Agreement via Generic Split KEM Construction". In: *Proceedings of the 10th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP*. INSTICC. SciTePress, 2024, pp. 594–608. ISBN: 978-989-758-683-5. DOI: 10.5220/0012454400003648.

[15] Keitaro Hashimoto et al. "An Efficient and Generic Construction for Signal's Handshake (X3DH): Post-Quantum, State Leakage Secure, and Deniable". In: *PKC 2021, Part II*. Ed. by Juan Garay. Vol. 12711. LNCS. Springer, Heidelberg, May 2021, pp. 410–440. DOI: 10.1007/978-3-030-75248-4_15.

[16] Keitaro Hashimoto et al. "An Efficient and Generic Construction for Signal's Handshake (X3DH): Post-quantum, State Leakage Secure, and Deniable". In: *Journal of Cryptology* 35.3 (July 2022), p. 17. DOI: 10.1007/s00145-022-09427-1.

[17] Loïs Huguenin-Dumittan and Serge Vaudenay. "On IND-qCCA Security in the ROM and Its Applications - CPA Security Is Sufficient for TLS 1.3". In: *EUROCRYPT 2022, Part III*. Ed. by Orr Dunkelman and Stefan Dziembowski. Vol. 13277. LNCS. Springer, Heidelberg, May 2022, pp. 613–642. DOI: 10.1007/978-3-031-07082-2_22.

[18] Duhyeong Kim et al. "Toward Practical Lattice-Based Proof of Knowledge from Hint-MLWE". In: *CRYPTO 2023, Part V*. Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14085. LNCS. Springer, Heidelberg, Aug. 2023, pp. 549–580. DOI: 10.1007/978-3-031-38554-4_18.

[19] Ehren Kret and Rolfe Schmidt. *The PQXDH Key Agreement Protocol*. Protocol specification. Oct. 18, 2023. URL: https://signal.org/docs/specifications/pqxdh/.

[20] Xingye Lu, Man Ho Au, and Zhenfei Zhang. "Raptor: A Practical Lattice-Based (Linkable) Ring Signature". In: *ACNS 19*. Ed. by Robert H. Deng et al. Vol. 11464. LNCS. Springer, Heidelberg, June 2019, pp. 110–130. DOI: 10.1007/978-3-030-21568-2_6.

[21] Vadim Lyubashevsky. "Lattice Signatures without Trapdoors". In: *EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. LNCS. Springer, Heidelberg, Apr. 2012, pp. 738–755. DOI: 10.1007/978-3-642-29011-4_43.

[22] Vadim Lyubashevsky et al. *CRYSTALS-DILITHIUM*. Tech. rep. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. National Institute of Standards and Technology, 2022.

[23] Moxie Marlinspike and Trevor Perrin. "The X3DH Key Agreement Protocol". In: *Open Whisper Systems* 283 (2016).

[24] D. Micciancio and O. Regev. "Worst-case to average-case reductions based on Gaussian measures". In: *45th Annual IEEE Symposium on Foundations of Computer Science*. 2004, pp. 372–381. DOI: 10.1109/FOCS.2004.72.

[25] Michael Naehrig et al. *FrodoKEM*. Tech. rep. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions. National Institute of Standards and Technology, 2020.

[26] Chris Peikert. "An Efficient and Parallel Gaussian Sampler for Lattices". In: *CRYPTO 2010*. Ed. by Tal Rabin. Vol. 6223. LNCS. Springer, Heidelberg, Aug. 2010, pp. 80–97. DOI: 10.1007/978-3-642-14623-7_5.

[27] Chris Peikert. "Lattice Cryptography for the Internet". In: *Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014*. Ed. by Michele Mosca. Springer, Heidelberg, Oct. 2014, pp. 197–219. DOI: 10.1007/978-3-319-11659-4_12.

[28] Thomas Prest et al. *FALCON*. Tech. rep. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. National Institute of Standards and Technology, 2022.

[29] Nathan Reitinger et al. "Is Cryptographic Deniability Sufficientf Non-Expert Perceptions of Deniability in Secure Messaging". In: *2023 IEEE Symposium on Security and Privacy (SP)*. 2023, pp. 274–292. DOI: 10.1109/SP46215.2023.10179361.

[30] Peter Schwabe et al. *CRYSTALS-KYBER*. Tech. rep. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. National Institute of Standards and Technology, 2022.

[31] Nihal Vatandas et al. "On the Cryptographic Deniability of the Signal Protocol". In: *ACNS 20, Part II*. Ed. by Mauro Conti et al. Vol. 12147. LNCS. Springer, Heidelberg, Oct. 2020, pp. 188–209. DOI: 10.1007/978-3-030-57878-7_10.

[32] Tarun Kumar Yadav, Devashish Gosain, and Kent Seamons. "Cryptographic deniability: a multi-perspective study of user perceptions and expectations". In: *Proceedings of the 32nd USENIX Conference on Security Symposium*. SEC '23. Anaheim, CA, USA: USENIX Association, 2023. ISBN: 978-1-939133-37-3.

[33] Tsz Hon Yuen et al. "DualRing: Generic Construction of Ring Signatures with Efficient Instantiations". In: *CRYPTO 2021, Part I*. Ed. by Tal Malkin and Chris Peikert. Vol. 12825. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 251–281. DOI: 10.1007/978-3-030-84242-0_10.

A Detailed parameters

This section provides the detailed parameter sets for Sparrow-KEM, with concrete values for the parameters of our construction, including the ring dimension, standard deviations and modulus q.

We provide the full parameter sets for our proved instantiation in Table 5, as well as parameter sets for our heuristic instantiation in Table 6.

Type	Asym	Sym
κ	128	128
n	64	1024
(k, ℓ)	$(18, 18)$	$(1, 1)$
q	261761	4188161
σ_{A}	2	4
σ_{B}	2	4
σ_y	2^7	2^9
B	3	5
ctbits	64	32
Deniability	148/127	91/78
OW-CPA	194/166	182/156
Decaps-OW-CPA (entropy)	128	128
Decaps-OW-CPA (MLWE)	130/102	112/91
$\lvert\mathsf{pk}_{\mathsf{A}}\rvert$	2 592B	2 816B
$\lvert\mathsf{pk}_{\mathsf{B}}\rvert$	2 592B	2 816B
$\lvert\mathsf{ct}\rvert$	40B	36B

Table 5: Proven parameter sets for Sparrow-KEM, aiming for NIST security level I. We provide NTT-friendly modulos q. We indicate the core-SVP hardness [1] for the deniability, OW-CPA, and Decaps-OW-CPA properties (hardness classical/quantum), minus the dimensions for free [10]. The core-SVP methodology ignores several polynomial factors, and is often used in the literature to compare lattice-based constructions. As a comparison, Dilithium-II [22] achieves 112/98 bits of core-SVP hardness.

Scheme	Sparrow-KEM		FrodoKex+
	Asym	Sym	
κ	128	128	
n	128	1024	
(k, ℓ)	$(7, 7)$	$(1, 1)$	
q	260609	120833	
σ_{A}	4	2	
σ_{B}	4	2	
σ_y	2^9	2^7	
B	2	2	
ctbits	128	128	
Deniability	123/106	113/97	
OW-CPA	164/140	182/156	
Decaps-OW-CPA (entropy)	128	128	
Decaps-OW-CPA (MLWE heur.)	135/115	154/132	
$\lvert\mathsf{pk}_{\mathsf{A}}\rvert$	2 016B	2 176B	21 300B
$\lvert\mathsf{pk}_{\mathsf{B}}\rvert$	2 016B	2 176B	21 300B
$\lvert\mathsf{ct}\rvert$	48B	48B	136B

Table 6: Heuristic parameter sets for Sparrow-KEM, plus sizes of FrodoKex+ [8] for comparison. We provide NTT-friendly modulos q. We indicate the core-SVP hardness for the deniability, OW-CPA, and Decaps-OW-CPA properties (hardness classical/quantum), which is a metric ignoring several polynomial factors. See the caption of Table 5 for details.

B Spectral norm of a gaussian vector

For completeness, we now recall the proof of Lemma 3, which was communicated to us by a private communication, and comes from another paper under review. We do not claim ownership of this result.

They complete their proof with an heuristic extending a norm bound result on lattices to a more complex structure. Although it is unclear how to fully formally prove it, it appears like a rather weak heuristic, and they also verify that the finalized lemma corresponds to the behavior observed in practice.

We first introduce some new notations to prove this result.

Recall that we consider polynomials in the cyclotomic ring $\mathcal{R} = \mathbb{Z}[x]/(x^n + 1)$ of degree n. Multiplications in this ring can be represented by matrices. Given a polynomial $a = \sum_{i\in[n]} a_i \cdot x^i \in \mathcal{R}$, we define the anti-circulant matrix $\mathcal{M}(a)$:

$$\mathcal{M}(a) = \begin{bmatrix} a_0 & -a_{n-1} & -a_{n-2} & \dots & -a_1 \\ a_1 & a_0 & -a_{n-1} & & -a_2 \\ a_2 & a_1 & a_0 & & -a_3 \\ \vdots & & & \ddots & \vdots \\ a_n & a_{n-1} & a_{n-2} & \dots & a_0 \end{bmatrix}$$

Note that $\mathcal{M}(a)$ is the matrix representation of the endomorphism $b \in \mathcal{R} \mapsto a \cdot b$ in the basis $\{1, x, ..., x^{n-1}\}$. We can additionally extend this notation to the operator $\varphi_{\mathbf{a}} : \mathbf{b} = \begin{bmatrix} b_0 \\ \vdots \\ b_{k-1} \end{bmatrix} \in \mathcal{R}^k \mapsto \sum_{i\in[k]} a_i \cdot b_i$, for some $\mathbf{a} = \begin{bmatrix} a_0 \\ \vdots \\ a_{k-1} \end{bmatrix} \in \mathcal{R}^k$, by defining $\mathcal{M}(\mathbf{a}) = \begin{bmatrix} \mathcal{M}(a_0) & \dots & \mathcal{M}(a_{k-1}) \end{bmatrix}$.

The spectral norm of a vector $\mathbf{a} \in \mathcal{R}^k$ is defined as the spectral norm of the operator $\varphi_{\mathbf{a}}$, which can equivalently be expressed using the matrix $\mathcal{M}(\mathbf{a})$: $s_1(\mathbf{a}) \coloneqq \mathcal{M}(\mathbf{a})$.

Our goal is to bound $s_1(\mathbf{a})$ when $\mathbf{a}$ is sampled from a Gaussian distribution.

The proof strategy relies on a change of basis using the complex roots of $x^n + 1$, noted $\Omega = \{\zeta_i, i \in [n]\}$. We also introduce the Vandermonde matrix $\mathbf{V} \coloneqq \left[\zeta_i^j\right]_{i,j\in[n]}$ associated to Ω. Recall that $\mathbf{V}^t \cdot \mathbf{V} = \mathbf{V} \cdot \mathbf{V}^t = n\mathbf{I}_n$.

The first useful lemma reexpresses $s_1(\mathbf{a})$ using the root of unity $(\zeta_i)_i$.

Lemma 4. *Let $\mathbf{a} = (a_i)_{i\in[k]} \in \mathcal{R}^k$. We have*

$$s_1(\mathbf{a}) = \max_{\zeta\in\Omega} \sqrt{\sum_{i\in[k]} |a_i(\zeta)|^2}$$

Proof. Observe for a single polynomial $a \in \mathcal{R}$ that we can diagonalize $\mathcal{M}(a)$ using $\mathbf{V}$: $\mathbf{U}^* \cdot \mathcal{M}(a) \cdot \mathbf{U} = \mathsf{Diag}(a(\zeta))_{\zeta\in\Omega}$, where $\mathbf{U} = \frac{1}{\sqrt{n}}\mathbf{V}$ and $\mathbf{U}^*$ is the conjugate transpose of $\mathbf{U}$.

Then, for a vector $\mathbf{a} = \begin{bmatrix} a_0 \\ \vdots \\ a_{k-1} \end{bmatrix} \in \mathcal{R}^k$, we observe that $\mathcal{M}(\mathbf{a}) \cdot \mathcal{M}(\mathbf{a})^t = \sum_{i\in[k]} \mathcal{M}(a_i) \cdot \mathcal{M}(a_i)^t$. We can diagonalize it as before:

$$\begin{aligned} \mathbf{U}^* \cdot \mathcal{M}(\mathbf{a}) \cdot \mathcal{M}(\mathbf{a})^t \cdot \mathbf{U} &= \sum_{i\in[k]} \mathsf{Diag}(|a_i(\zeta)|^2)_{\zeta\in\Omega} \\ &= \mathsf{Diag}\left(\sum_{i\in[k]} |a_i(\zeta)|^2\right)_{\zeta\in\Omega} \end{aligned}$$

Since the spectral norm of $\mathcal{M}(\mathbf{a})$ is the largest eigenvalue of $\mathcal{M}(\mathbf{a}) \cdot \mathcal{M}(\mathbf{a})^t$, this concludes the proof. □

Before proving the main theorem, we introduce useful lemmas.

Lemma 5 (Lemma 4.4 in [21], extended). *For any $\tau > 1$, and any lattice $\Delta \subseteq \mathbb{R}^n$,*

$$\Pr\left[\|\mathbf{z}\| > \tau\sigma\sqrt{n} \mid \mathbf{z} \leftarrow \mathcal{D}_{\Delta,\sigma}\right] \leqslant C^n$$

where $C = \tau \cdot e^{\frac{1}{2}(1-\tau^2)} < 1$. In particular, if $(\tau \geqslant 1 + \frac{\kappa \log 2}{3n})$ or if $(\tau \geqslant 1 + \sqrt{\frac{4\kappa \log 2}{n}})$, then $C^n \leqslant 2^{-\kappa}$.

Lemma 5 is easily derived from the proof of Lemma 4.4 in [21], by observing that [2] applies to any lattice and not only $\mathbb{Z}^n$.

Lemma 6 (Lemma 2.3 of [13]). *For any lattice $\Delta \subseteq \mathbb{R}^n$, matrix $T \in \mathbb{R}^{k \times n}$, and real $\sigma > 0$. If T is injective on Δ, then*

$$T \cdot \mathcal{D}_{\Delta,\sigma} = \mathcal{D}_{T\cdot\Delta,\sqrt{TT^t}\cdot\sigma}$$

We now have all the components to prove Lemma 3.

of Lemma 3. Consider a vector $\mathbf{a} \leftarrow \mathcal{D}_{\mathcal{R}^k,\sigma}$.

Recall from Lemma 4 that $s_1(\mathbf{a}) = \max_{\zeta\in\Omega} \sqrt{\sum_{i\in[k]} |a_i(\zeta)|^2}$.

Fix some $\zeta_j \in \Omega$, we can express $\sqrt{\sum_{i\in[k]} |a_i(\zeta_j)|^2}$ as follow:

$$\begin{bmatrix} a_0(\zeta_j) \\ \vdots \\ a_{k-1}(\zeta_j) \end{bmatrix} = \mathbf{T} \cdot \mathbf{a}$$

where $\mathbf{T} = \mathbb{Z}^k \bigotimes \begin{bmatrix} 1 & \zeta_j & \dots & \zeta_j^{n-1} \end{bmatrix} \in \mathbb{C}^{nk\times nk}$.

Note $\mathbf{T}' \coloneqq \begin{bmatrix} \mathsf{Re}(\mathbf{T}) \\ \mathsf{Im}(\mathbf{T}) \end{bmatrix}$. Then the vector $\mathbf{T}' \cdot \mathbf{a}$ has as coordinates the real and imaginary values of $a_0(\zeta_j), ..., a_{k-1}(\zeta_j)$. In particular, $|p(\mathbf{T}' \cdot \mathbf{a})|^2 = \sum_{i\in[k]} |a_i(\zeta_j)|^2$.

We now would like to bound $|\mathbf{T}' \cdot \mathbf{a}|$. This is where some heuristics kick in.

We heuristically assume that Lemma 6 applies on $\mathbf{T}' \cdot \mathbf{a}$ as if $\mathbf{T}'$ was injective. Then, we would have that $\mathbf{T}' \cdot \mathbf{a}$ follows the Gaussian distribution $\mathcal{D}_{\mathbf{T}'\cdot\mathbb{Z}^{nk},\sqrt{\mathbf{T}'\mathbf{T}'^t}\sigma}$. Interestingly, $\mathbf{T}' \cdot \mathbf{T}'^t = \frac{n}{2} \cdot \mathbf{I}_{2k}$, and thus $\mathbf{T}' \cdot \mathbf{a}$ heuristically follows $\mathcal{D}_{\mathbf{T}'\cdot\mathbb{Z}^{nk},\sqrt{n/2}\sigma}$.

In a second step, we assume that $\mathcal{D}_{\mathbf{T}'\cdot\mathbb{Z}^{nk},\sqrt{n/2}\sigma}$ behaves like a discrete Gaussian over an arbitrary lattice, even though $\mathbf{T}' \cdot \mathbb{Z}^{nk}$ is not a lattice. Heuristically applying Lemma 5, we obtain that $\|\mathbf{T} \cdot \mathbf{a}\| \leqslant \tau \cdot \sqrt{n/2}\sigma \cdot \sqrt{2k} = \tau \cdot \sigma \cdot \sqrt{nk}$ except with probability C^{2k}. Applying the union bound for the $n/2$ $\zeta \in \Omega$ with positive imaginary part concludes the proof. □