Wyner-Ziv reconciliation for key exchange based on Ring-LWE

Charbel Saliba, Laura Luzzi, Cong Ling

Wyner-Ziv reconciliation for key exchange based on Ring-LWE

Charbel Saliba and Laura Luzzi
ETIS UMR 8051, Université Paris Seine,
Université Cergy-Pontoise, ENSEA, CNRS,
Cergy, France
Email: {charbel.saliba, laura.luzzi}@ensea.fr

Cong Ling
Department of Electrical
and Electronic Engineering
Imperial College London, U.K.
Email: cling@ieee.org

***Abstract*—We consider a key encapsulation mechanism (KEM) based on ring-LWE where reconciliation is performed on an N-dimensional lattice using Wyner-Ziv coding. More precisely, we consider Barnes-Wall lattices and use Micciancio and Nicolosi's bounded distance decoder with polynomial complexity $\mathcal{O}(N\log^2(N))$. We show that in the asymptotic regime for large N, the achievable key rate is $\Theta(\log N)$ bits per dimension, while the error probability $P_e \approx \mathcal{O}(e^{-N\epsilon})$. Unlike previous works, our scheme does not require a dither.**

I. Introduction

Over the past few years, there have been many attractive developments in lattice-based cryptographic protocols, whose security is based on *worst-case* hardness assumptions, and which are conjectured to be secure against *quantum* attacks. Thus, lattice-based primitives are a promising candidate to replace constructions based on number theoretic assumptions like RSA [1] or Diffie-Hellman [2] that are currently in use.

One of the most versatile primitives for the design of provably secure cryptographic protocols is the *learning with errors* (LWE) problem introduced by Regev [3]. For instance it can serve for IND-CPA (Indistinguishability under chosen-plaintext attack) [3] and IND-CCA (Indistinguishability under chosen-ciphertext attack) public key encryption [4]. A structured variant of LWE, the decision *ring learning with errors* (R-LWE) was proposed in [5] by Lyubashevsky *et al.* to allow more compact representations. Cryptographic applications of R-LWE include fast encryption [5] and fast homomorphic encryption [6]. Solving R-LWE is at least as hard as solving approximate SIVP on ideal lattices.

In [7], Peikert introduced an efficient lattice-based *key encapsulation mechanism* (KEM) that allows two parties to share an ephemeral key that is useful for secret communications, featuring a low bandwidth *reconciliation* technique that aims to reach exact agreement on the shared key. A practical implementation of Peikert's protocol called NewHope was proposed in [8] as a candidate to the NIST challenge on post-quantum cryptography. In [7] and [8], although key generation is performed using N-dimensional lattices, the reconciliation step uses 1-dimensional and 4-dimensional lattices respectively[1].

[1]In fact, the latest implementation of the NewHope algorithm does not use reconciliation [9].

In this paper, we consider a more general framework for KEM based on ring-LWE, that does not require a dither, and where reconciliation is done directly on the N-dimensional lattice using Wyner-Ziv coding.

More precisely, we consider Barnes-Wall lattices [10] and use Micciancio and Nicolosi's BDD decoder with polynomial complexity [11] for the reconciliation step. In particular, we prove that this decoder is linear. This result is required for our security proof and may also be of independent interest. In the asymptotic regime for large N, we show that this technique can generate $\Theta(\log N)$ bits of key per dimension. This improves upon [7] and [8] where the key rates are 1 bit and $1/4$ bits per dimension respectively. Moreover, our scheme achieves exponentially small error probability $P_e \approx \mathcal{O}(e^{-N\epsilon})$, in particular $P_e = 2^{-1675}$ when $N = 1024$. Although current recommendations are to keep the error probability smaller than 2^{-128}, this may be too conservative when transforming an IND-CPA secure encryption scheme into an IND-CCA secure one using the Fujisaki-Okamoto transform [12]. A smaller error probability is desirable to prevent leakage of information from decryption failure attacks [13].

Organization: This paper is organized as follows. In Section II we provide basic definitions about cyclotomic fields, lattices, etc. In Section III we present the Barnes-Wall lattice with some of its properties. In Section IV, we introduce our key generation algorithm. In Section V and VI, we provide a proof that the error probability is small, and that our scheme is IND-CPA secure respectively.

II. Preliminaries

In this section, we introduce the mathematical tools we use to describe and analyze our proposed scheme.

We write $f(N) = \omega(g(N))$ if $\lim_{N\to\infty}(f(N)/g(N)) = \infty$, and $f(N) = \Theta(g(N))$ if $f(N) = O(g(N))$ and $g(N) = O(f(N))$. Finally a variant of O notation that "ignores" logarithmic factors: $f(N) = \tilde{O}(g(N))$, equivalent to $f(N) = O\left(g(N)\cdot\log^k(g(N))\right)$ for some integer k.

A. Lattices and Algebraic number theory

Lattice definitions: First of all, we define the *space H* as follows: when $m \geq 2$ and $N = \varphi(m)$ with φ the Euler's

totient function, let

$$H = \{\mathbf{x} \in \mathbb{C}^N; \quad x_i = \overline{x_{m-i}}, \forall i \in \mathbb{Z}_m^*\} \subsetneq \mathbb{C}^N.$$

Note that H is a proper $\mathbb{R}$-subspace of $\mathbb{C}^N$ and is isomorphic to $\mathbb{R}^N$ as an inner product space.

For our purposes, a *lattice* is a real full-rank discrete additive subgroup $\Lambda \subseteq H$. Any lattice is generated as the set of all integer linear combinations of N linearly independent basis vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_N\}$ in H as $\Lambda = \Lambda(\mathbf{B}) = \left\{\mathbf{B} \cdot \mathbf{z};\ \mathbf{z} \in \mathbb{Z}^N\right\}$. A fundamental cell $\mathcal{P}_0$ of Λ is a bounded set, which, when shifted by the lattice points, generates a partition $\mathcal{P} = \{\mathcal{P}_\lambda\}$ of $\mathbb{R}^N$. For a fundamental cell $\mathcal{P}_0$, any point $\mathbf{x} \in \mathbb{R}^N$ can be uniquely expressed as a sum

$$\mathbf{x} = \lambda + \mathbf{x}_e, \quad \text{where } \lambda \in \Lambda \text{ and } \mathbf{x}_e \in \mathcal{P}_0.$$

We write $\lambda = Q_{\mathcal{P}(\Lambda)}(\mathbf{x})$ and $\mathbf{x}_e = \mathbf{x} \bmod \mathcal{P}_0 = \mathbf{x} - Q_{\mathcal{P}(\Lambda)}(\mathbf{x})$. We will use implicitly in our proofs the fact that $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^N$, and $\forall \lambda \in \Lambda$, $(\mathbf{x} \bmod \Lambda + \mathbf{y}) \bmod \Lambda = (\mathbf{x} + \mathbf{y}) \bmod \Lambda$ as well as $(\mathbf{x} + \lambda) \bmod \Lambda = \mathbf{x} \bmod \Lambda$.
Given a lattice Λ with basis $\mathbf{B}$ and a vector $\mathbf{t}$ such that $\mathrm{dist}(\mathbf{t}, \mathbf{B}) \leq \frac{1}{2} d_{\min}(\Lambda)$, the *bounded distance decoding* problem is to find the lattice vector $\mathbf{v} \in \Lambda$ closest to $\mathbf{t}$.

Lemma 1. Let $\Lambda' \subset \Lambda$ and $\lambda \in \Lambda$; then $\pi : \Lambda/\Lambda' \to \Lambda/\Lambda'$ defined as $\pi(\mathbf{v}) = (\mathbf{v} + \lambda) \bmod \Lambda'$ is a permutation of Λ/Λ'.

Cyclotomic fields and the canonical embedding: For an integer $m \geq 1$, the m^{th} *cyclotomic number field* is the extension $K = \mathbb{Q}(\xi_m)$ with degree $N = \varphi(m)$, where ξ_m is any m^{th} primitive root of unity. We denote the *ring of integers* $\mathcal{O}_K$ of K by R, its *co-different* by $R^\vee$, and define $R_q = R/qR$ for any integer $q \geq 1$. In the same manner we can define $R_q^\vee$. Note that R_q is isomorphic to $R_q^\vee$ by an isomorphism θ [5, Lemma 2.15].

Now we describe the *embedding* of a cyclotomic number field, which induces a "canonical" geometry on it. $\mathbb{Q}(\xi_m)$ will has exactly N injective ring homomorphisms $\sigma_i : K \mapsto \mathbb{C}$, and we can define the *canonical embedding* $\sigma : K \mapsto \mathbb{C}^N$ as

$$\sigma(a) = (\sigma_1(a), \sigma_2(a), \ldots, \sigma_N(a)) \in H \subsetneq \mathbb{C}^N.$$

This is a ring homomorphism from K to H, where multiplication and addition in the latter are both component-wise. We define norms and other geometric quantities on K simply by identifying field elements $a \in K$ with their canonical embeddings $\sigma(a) \in H$, e.g., the l_2 norm is $\|a\|_2 = \|\sigma(a)\|_2$.

When dealing with cyclotomic number fields, note that if m is a power of 2 with $N = \varphi(m)$, then $R^\vee = \frac{1}{N} R$, and $\sigma(R) = \sqrt{N} \cdot \Phi \cdot \mathbb{Z}^N$ for some orthogonal matrix Φ [14].

B. Error Distribution

Subgaussian vectors: A random vector X^N in $\mathbb{R}^N$ is *subgaussian* with parameter $s > 0$, if for any unit vector $\mathbf{u} \in \mathbb{R}^N$ and for any $t \in \mathbb{R}$,

$$\mathbb{E}\left[e^{2\pi t \langle \mathsf{X}^N, \mathbf{u} \rangle}\right] \leq e^{\pi t^2 s^2}.$$

As a consequence of Theorem 1 in [15], the following tail inequality holds.

Theorem 1. Let X^N be a subgaussian vector in $\mathbb{R}^N$ with parameter $s > 0$. Then $\forall \epsilon > 0$:

$$\mathbb{P}\left\{\|\mathsf{X}^N\| > \tfrac{s}{\sqrt{2\pi}} \sqrt{N} \cdot (\epsilon + 1)\right\} \leq e^{-N\epsilon^2/2}.$$

The following two propositions describe the sum and point-wise product behavior of subgaussians:

Proposition 1 ([16], Corollary 2.3)**.** Let $\mathsf{X}_1^N, \ldots, \mathsf{X}_k^N$ be independent subgaussian vectors over $\mathbb{R}^N$ with parameters s_i. Then $\sum \mathsf{X}_i^N$ is subgaussian with parameter $s = \left(\sum s_i^2\right)^{\frac{1}{2}}$.

Proposition 2 ([16], Claim 2.4)**.** Let X^N be a subgaussian vector in $\mathbb{R}^N$ of parameter s, and Y^N another random vector. Then the point-wise multiplication vector $\mathsf{Z}^N = \mathsf{X}^N \odot \mathsf{Y}^N = (\mathsf{X}_1\mathsf{Y}_1, \ldots, \mathsf{X}_N\mathsf{Y}_N)$ is subgaussian of parameter $\|\mathsf{Y}\|_2 \cdot s$.

Gaussian-like distribution: When dealing with ring-LWE defined below, we work with a Gaussian-like error distribution over the number field K. We first define the N-dimensional i.i.d. Gaussian distribution $D_{r'}$ with zero mean and covariance r'. Then we define the Gaussian distribution ψ over $K \otimes_{\mathbb{Q}} \mathbb{R}$ to output an element $a \in K \otimes_{\mathbb{Q}} \mathbb{R}$ for which $\sigma(a) \in H$ has Gaussian distribution $D_{r'}$ with parameter r'. In our application, we discretize ψ to $R^\vee$ using coordinate-wise randomized rounding [16] and denote the resulting distribution by $\chi = \lfloor \psi \rceil_{R^\vee}$.

Proposition 3 ([16], Lemma 8.2)**.** If ψ is a continuous Gaussian with parameter $r' \geq 1$, and we use coordinate-wise randomized rounding, then $\chi = \lfloor \psi \rceil_{R^\vee}$ is subgaussian with parameter $r = \sqrt{r'^2 + 2\pi \cdot \mathrm{rad}(m)/m} = \mathcal{O}(r')$, where $\mathrm{rad}(m)$ is the product of all distinct primes dividing m.

C. Ring-LWE

A function $f(N)$ is *negligible* if $f(N) = o(N^{-c})$ for any constant $c \geq 0$. Two ensembles $\{\mathsf{X}_N\}_{N \in \mathbb{N}}$ and $\{\mathsf{Y}_N\}_{N \in \mathbb{N}}$ are *computationally indistinguishable* if for all efficient distinguisher algorithms $\mathcal{D}$, $|\mathbb{P}\{\mathcal{D}(\mathsf{X}_N) = 1\} - \mathbb{P}\{\mathcal{D}(\mathsf{Y}_N) = 1\}|$ is negligible in N.
We define the notion of *key encapsulation mechanism* (KEM). Following [7], a KEM with ciphertext space $\mathcal{C}$ and (finite) key space $\mathcal{K}$ is given by efficient algorithms Setup, Gen, Encaps and Decaps, having the following structure:

- $\mathsf{Setup}()$ outputs a public parameter pp.
- $\mathsf{Gen}(pp)$ outputs a public encapsulation key pk and secret decapsulation key sk.
- $\mathsf{Encaps}(pp;\ pk)$ outputs a ciphertext $c \in \mathcal{C}$ and a key $k \in \mathcal{K}$.
- $\mathsf{Decaps}(sk;\ c)$ outputs some $k \in \mathcal{K} \cup \{\text{error term}\}$.

A KEM satisfies IND-CPA security, if the outputs of the following "real" and "ideal" games are computationally indistinguishable:

Real Game	**Ideal Game**
$pp \leftarrow \mathsf{Setup}()$	$pp \leftarrow \mathsf{Setup}()$
$(pk, sk) \leftarrow \mathsf{Gen}(pp)$	$(pk, sk) \leftarrow \mathsf{Gen}(pp)$
$(c, k) \leftarrow \mathsf{Encaps}(pp, pk)$	$(c, k) \leftarrow \mathsf{Encaps}(pp, pk)$
	$k^* \leftarrow \mathcal{K}$
$\mathrm{Output}(pp, pk, c, k)$	$\mathrm{Output}(pp, pk, c, k^*)$

Ring-LWE: We state the ring-LWE problem in its discretized form. First of all, let's define the ring-LWE distribution:

Definition 1. For a distribution χ on $R^\vee$ and $\mathbf{s} \xleftarrow{\$} \chi$, a sample from the ring-LWE distribution $A_{\mathbf{s},\chi}$ over $R_q \times R_q^\vee$ is generated by choosing $\mathbf{a} \leftarrow R_q$ uniformly at random, choosing $\mathbf{e} \leftarrow \chi$, and outputting $(\mathbf{a}, \mathbf{b} = \mathbf{a} \cdot \mathbf{s} + \mathbf{e}) \in R_q \times R_q^\vee$.

Definition 2 (Ring-LWE, Decision)**.** The *decision* version of the ring-LWE problem, denoted R-DLWE$_{q,\chi}$, is to distinguish with non-negligible advantage between independent samples from $A_{\mathbf{s},\chi}$, where $\mathbf{s} \xleftarrow{\$} \chi$ is chosen once and for all, and the same number of uniformly random and independent samples from $R_q \times R_q^\vee$.

Theorem 2 ([16], Theorem 2.22)**.** Let R be the mth cyclotomic ring, having dimension $N = \varphi(m)$. Let $\alpha = \alpha(N) < \sqrt{\log N/N}$, and let $q = q(N) = 1 \bmod m$ be a poly(N)-bounded prime such that $\alpha q \geq \omega(\sqrt{\log N})$. There is a poly$(N)$-time quantum reduction from $\tilde{O}(\sqrt{N}/\alpha)$-approximate SIVP (*or* SVP) on ideal lattices in R to solving R-DLWE$_{q,\psi}$ given only ℓ samples, where ψ is the Gaussian distribution $D_{\xi q}$ for $\xi = \alpha \cdot (N\ell/\log(N\ell))^{1/4}$.

The following result extends the hardness guarantees to the case of discrete error. We make use of what is called a *valid discretization* from Section 2.4.2 in [16]:

Theorem 3 ([16], Lemma 2.24)**.** Let $\lfloor \cdot \rceil$ be a coordinate-wise randomized rounding to $R^\vee$. If R-DLWE$_{q,\psi}$ is hard given ℓ samples, then so is the variant of R-DLWE$_{q,\chi}$ in which the secret is sampled from χ given $\ell - 1$ samples.

Remark 1. To apply Theorem 3 with two samples, we let $\ell = 3$. Hence ψ is a continuous Gaussian with parameter $r' = q \cdot \alpha \cdot (3N/\log(3N))^{1/4}$.

III. Barnes-Wall lattices and the Micciancio-Nicolosi BDD decoder

The Barnes-Wall lattice BW^n is an $N = 2^n$ dimensional lattice over the Gaussian integers $\mathbb{G} = \mathbb{Z}[i] \cong \mathbb{Z}^2$ [11, 10]. Note that BW^n can be seen as a real lattice contained in $\mathbb{Z}^{2N} = \mathbb{Z}^{2^{n+1}}$. Moreover, $d_{\min}(BW^n) = \sqrt{N}$ and $\mathrm{Vol}(BW^n) = \sqrt{N^N}$.

Micciancio and Nicolosi [11] give a polynomial time algorithm to solve the bounded distance decoding (BDD) for Barnes-Wall lattices: given a vector $\mathbf{s} \in \mathbb{C}^N$ within distance $d_{\min}/2 = \sqrt{N}/2$ from some lattice point λ in BW^n, find λ. This algorithm called ParBW has complexity $\mathcal{O}\left(N \log^2 N\right)$ and we can prove that it is linear in the following sense:

Theorem 4. Let $\mathbf{s} \in \mathbb{C}^N$. For a fixed $\lambda \in BW^n$, we have

$$\textsc{ParBW}(\lambda + \mathbf{s}) = \lambda + \textsc{ParBW}(\mathbf{s}).$$

This means that the operation ParBW induces a partition of $\mathbb{C}^N \cong \mathbb{Z}^{2N}$ into fundamental cells. The proof of Theorem 4 can be found in the Appendix.

We can also scale the Barnes-Wall lattice by an $N \times N$ matrix T to obtain $T \cdot BW^n$. For invertible matrix T and $\mathbf{s} \in \mathbb{C}^N$ we define the operation $\textsc{ParBW}_T$ as:

$$\textsc{ParBW}_T(\mathbf{s}) = T \cdot \textsc{ParBW}(T^{-1} \cdot \mathbf{s}),$$

and $\mathbf{x} \bmod (T \cdot BW^n) = \mathbf{x} - \textsc{ParBW}_T(\mathbf{x})$. It is not hard to prove that for $\lambda' \in T \cdot BW^n$ and $\mathbf{s} \in \mathbb{C}^N$, $\textsc{ParBW}_T(\lambda' + \mathbf{s}) = \lambda' + \textsc{ParBW}_T(\mathbf{s})$, and $(\mathbf{s} + \lambda') \bmod (T \cdot BW^n) = \mathbf{s} \bmod (T \cdot BW^n)$.

Proposition 4. For any $k \geq \left\lfloor \frac{n+1}{2} \right\rfloor$, we have that $2^k \mathbb{Z}^{2N} \subseteq BW^n \subseteq \mathbb{Z}^{2N}$, where $N = 2^n$.

IV. Key generation algorithm

We give here the key generation algorithm below between Alice and Bob.

Parameters are q; $N = [K : \mathbb{Q}]$ and error distribution χ on $R^\vee$		
Alice (Server)		**Bob (Client)**
$\mathbf{a} \xleftarrow{\$} R_q$		
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi$		$\mathbf{s}', \mathbf{e}', \mathbf{e}'' \xleftarrow{\$} \chi$
$\mathbf{b} := \mathbf{as} + \mathbf{e} \in R_q^\vee$	$\xrightarrow{(\mathbf{b},\mathbf{a})}$	
		$\mathbf{u} := \mathbf{as}' + \mathbf{e}'$
		$\mathbf{v} := \theta^{-1}(\mathbf{b})\mathbf{s}' + \mathbf{e}''$
$\mathbf{v}' := \theta^{-1}(\mathbf{u})\,\mathbf{s}$	$\xleftarrow{(\mathbf{u},\mathbf{r})}$	$\mathbf{r} = Q_{\Lambda_1}(\mathbf{v}) \bmod \Lambda_2$
$\hat{\mathbf{k}} = Q_{\Lambda_2}(\mathbf{v}' - \mathbf{r}) \bmod \Lambda_3$		$\mathbf{k} = Q_{\Lambda_2}(\mathbf{v} - \mathbf{r}) \bmod \Lambda_3$

TABLE I
Key generation algorithm

We start by considering the lattice $\Lambda_0 = \sigma(R^\vee) = \frac{1}{\sqrt{N}} \cdot \Phi \cdot \mathbb{Z}^N$, a scaled rotation of $\mathbb{Z}^N$ where $N = \varphi(m)$ with m a power of 2. After that, $\Lambda_3 = q\Lambda_0 = \frac{1}{\sqrt{N}} \cdot \Phi \cdot q\mathbb{Z}^N$, thence Λ_0/Λ_3 is identified to $\sigma(R_q^\vee)$. Note that σ induces an isomorphism between the additive quotient groups $R_q^\vee$ and $\Lambda_0/q\Lambda_0 = \Lambda_0/\Lambda_3$. With slight abuse of notation, in the rest of the paper we identify the two quotient groups. For the remaining two lattices, we choose Λ_1 (quantization lattice) and Λ_2 (coding lattice) with partitions $\mathcal{P}_1, \mathcal{P}_2$ into fundamental sets such that the operation $Q_{\mathcal{P}_i(\Lambda_i)}$ can be done in polynomial time for $i = 1, 2$ and such that Q_{Λ_2} performs BDD, i.e. given $\mathbf{s} \in \mathbb{Z}^{2N}$ within distance $d_{\min}(\Lambda_2)/2$ from $\lambda \in \Lambda_2$, $Q_{\Lambda_2}(\mathbf{s}) = \lambda$. We will use the notation Q_{Λ_i} when there is no ambiguity about the chosen partition. Moreover, we impose that $\Lambda_2 \subseteq \Lambda_0$ and $\Lambda_3 \subseteq \Lambda_2 \subseteq \Lambda_1$.

The *reconciliation rate* og the protocol is $R_P = \frac{1}{N} \cdot \log_2\left(\frac{\mathrm{Vol}(\Lambda_2)}{\mathrm{Vol}(\Lambda_1)}\right)$, and the *key rate* $R_K = \frac{1}{N} \cdot \log_2\left(\frac{\mathrm{Vol}(\Lambda_3)}{\mathrm{Vol}(\Lambda_2)}\right)$.

We suppose that the error terms $\mathbf{e}, \mathbf{e}', \mathbf{e}''$, and the secret terms $\mathbf{s}, \mathbf{s}'$, are taken independently from the χ distribution on $R^\vee$, which is subgaussian with parameter r (see Proposition 3). We define the *modulus to noise ratio* as the quotient between the modulus q and the parameter r of the error distribution χ. A smaller modulus to noise ratio provides stronger concrete security against known attacks. Moreover, since all the exchanged messages are modulo q, the size of q affects the overhead of the protocol.

Referring to Table I, the KEM algorithm consists of the following steps:

- Setup() : Alice chooses a random element $\mathbf{a}$ from R_q and outputs $pp = \mathbf{a}$.
- Gen($\mathbf{a}$) : She then chooses $\mathbf{e}, \mathbf{s} \leftarrow \chi$ in $R_q^\vee$, computes $\mathbf{b} = \mathbf{a} \cdot \mathbf{s} + \mathbf{e}$, and outputs a public key $pk = \mathbf{b}$ and a secret key $sk = \mathbf{s}$.
- Encaps($pp = \mathbf{a}, pk = \mathbf{b}$) : Bob chooses independent $\mathbf{e}', \mathbf{e}'', \mathbf{s}' \leftarrow \chi$. He then computes $\mathbf{u} = \mathbf{a} \cdot \mathbf{s}' + \mathbf{e}' \in R_q^\vee$ and $\mathbf{v} = \theta^{-1}(\mathbf{b}) \cdot \mathbf{s}' + \mathbf{e}'' \in R_q^\vee$. He outputs $c = (\mathbf{u}, \mathbf{r}) \in \mathbb{R}_q^\vee \times \Lambda_1/\Lambda_2$ with

$$\mathbf{r} := \text{HelpRec}(\mathbf{v}) = Q_{\Lambda_1}(\mathbf{v}) \bmod \Lambda_2 \tag{1}$$

  and $\mathbf{k}$ in Λ_2/Λ_3 such that

$$\mathbf{k} := \text{Rec}(\mathbf{v}, \mathbf{r}) = Q_{\Lambda_2}(\mathbf{v} - \mathbf{r}) \bmod \Lambda_3. \tag{2}$$

- Decaps($sk = \mathbf{s}, c = (\mathbf{u}, \mathbf{r})$) : Alice computes $\mathbf{v}' := \theta^{-1}(\mathbf{u}) \cdot \mathbf{s}$ and outputs $\hat{\mathbf{k}} = \text{Rec}(\mathbf{v}', \mathbf{r})$.

Remark 2. This algorithm can essentially be seen as a generalization of the KEM in [7] and [8], where the reconciliation step is also lattice-based. For instance, in [8] the functions HelpRec and Rec can be written in the form (1) and (2) by taking $\Lambda_0 = \mathbb{Z}^{1024}$, and the product lattices $\Lambda_1 = (q\tilde{D}_4/2^r)^{256}$, $\Lambda_2 = (q\tilde{D}_4)^{256}$. Note that unlike [7, 8], a dither is not required in our algorithm.

a) Construction using Barnes-Wall lattices: For an explicit construction we choose $\Lambda_1 = \Lambda_0$ and $\Lambda_2 = T \cdot BW^{n-1}$, where $T = \beta \cdot \frac{1}{\sqrt{N}} \cdot \Phi$ and β a power of 2. By this choice, all the operations with Λ_2 in Table I can be deduced from Section III. The operation PARBW_T corresponds to a quantization operation $Q_{\Lambda_2,\mathcal{P}}$ induced by a partition $\mathcal{P}$ of the Barnes-Wall lattice: $\text{PARBW}_T = Q_{\Lambda_2,\mathcal{P}}$ (see Theorem 4). Since $\beta BW^{n-1} \subseteq BW^{n-1} \subseteq \mathbb{Z}^N$, we obtain that $\Lambda_2 \subseteq \Lambda_1 = \Lambda_0$. For the inclusion $\Lambda_3 \subseteq \Lambda_2$, we must have $q\mathbb{Z}^N \subseteq \beta BW^{n-1}$, or $\frac{q}{\beta}\mathbb{Z}^N \subseteq BW^{n-1}$. By Proposition 4, this is true when

$$q/\beta = 2^k, \text{with } k \geq \lfloor n/2 \rfloor. \tag{3}$$

Note that the key rate of the protocol is $\frac{1}{4}\log N - \frac{1}{4} \approx \Theta(\log N)$.

V. Error probability

Here we give a general estimation for the error probability $\mathbb{P}\{\mathbf{k} \neq \hat{\mathbf{k}}\}$, and then specialize to the case when Λ_2 is a Barnes-Wall lattice. We start by observing that $\mathbf{v} = \mathbf{ass}' + \mathbf{es}' + \mathbf{e}''$ and $\mathbf{v}' = \mathbf{ass}' + \mathbf{e}'\mathbf{s}$, therefore $\mathbf{v} = \mathbf{v}' + \bar{\mathbf{e}}$ with $\bar{\mathbf{e}} = \mathbf{es}' + \mathbf{e}'' - \mathbf{e}'\mathbf{s}$. Define the quantization error as $e_Q = \mathbf{v} - Q_{\Lambda_1}(\mathbf{v})$. Hence, $\mathbf{v} - \mathbf{r} = [e_Q + Q_{\Lambda_1}(\mathbf{v})] - [Q_{\Lambda_1}(\mathbf{v}) - Q_{\Lambda_2}(Q_{\Lambda_1}(\mathbf{v}))] = e_Q + Q_{\Lambda_2}(Q_{\Lambda_1}(\mathbf{v}))$. In the expressions of the shared keys we obtain

$$\mathbf{k} - \hat{\mathbf{k}} = [Q_{\Lambda_2}(e_Q) - Q_{\Lambda_2}(e_Q - \bar{\mathbf{e}})] \bmod \Lambda_3.$$

Note that $\mathbf{k} = \hat{\mathbf{k}}$ if $Q_{\Lambda_2}(e_Q) = 0$ and $Q_{\Lambda_2}(e_Q - \bar{\mathbf{e}}) = 0$.
To simplify the analysis we suppose from now on that $\Lambda_0 = \Lambda_1$ so that $e_Q = 0$. [2] Due to the BDD assumption for Q_{Λ_2}, we have that $\mathbf{k} = \hat{\mathbf{k}}$ if $\|\bar{\mathbf{e}}\| \leq \frac{1}{2} d_{\min}(\Lambda_2)$. Now we want to estimate

$$\mathbb{P}\left\{\|\bar{\mathbf{e}}\|_2 \geq \tfrac{d_{\min}(\Lambda_2)}{2}\right\} = \mathbb{P}\left\{\|\mathbf{es}' + \mathbf{e}'' - \mathbf{e}'\mathbf{s}\|_2 \geq \tfrac{d_{\min}(\Lambda_2)}{2}\right\}. \tag{4}$$

For any constant $c > 0$, by the law of total probability the term (4) can be bounded by

$$\mathbb{P}\{\|\mathbf{e}\|_2 > c\} + \mathbb{P}\{\|\mathbf{s}\|_2 > c\} + \mathbb{P}\left\{\|\bar{\mathbf{e}}\|_2 > \tfrac{d_{\min}(\Lambda_2)}{2} \;\middle|\; \begin{matrix}\|\mathbf{e}\|_2 \leq c, \\ \|\mathbf{s}\|_2 \leq c.\end{matrix}\right\} \tag{5}$$

Assuming that $\|\mathbf{e}\|_2 \leq c$, and $\mathbf{s}'$ is subgaussian with parameter r, then by Proposition 2, we can say that $\sigma(\mathbf{e}) \odot \sigma(\mathbf{s}')$ is subgaussian with parameter $\|\sigma(\mathbf{e})\|_2 \cdot r$; and so $\mathbf{es}'$ is subgaussian with parameter $c \cdot r$. Following the same argument, given that $\|\mathbf{s}\|_2 \leq c$ we get that $\mathbf{e}'\mathbf{s}$ is subgaussian with parameter $c \cdot r$. Since $\mathbf{e}''$ is subgaussian with parameter r, then under the condition that $\|\mathbf{e}\|_2 \leq c$ and $\|\mathbf{s}\|_2 \leq c$ we obtain using Proposition 1:

$$\bar{\mathbf{e}} \text{ is subgaussian with parameter } \bar{r} = r \cdot \sqrt{2c^2 + 1}.$$

Therefore, by Theorem 1 if we set $c = \frac{r}{\sqrt{2\pi}}\sqrt{N} \cdot (\epsilon + 1)$, and $\frac{d_{\min}(\Lambda_2)}{2} \geq \frac{\bar{r}}{\sqrt{2\pi}}\sqrt{N} \cdot (\epsilon + 1)$, then

$$\mathbb{P}\left\{\|\mathbf{es}' + \mathbf{e}'' - \mathbf{e}'\mathbf{s}\|_2 \geq \tfrac{d_{\min}(\Lambda_2)}{2}\right\} \leq 3 \cdot e^{-N\epsilon^2/2}. \tag{6}$$

Choose $\epsilon = \sqrt{2\pi} - 1$. The above conditions become $c = r\sqrt{N}$ and

$$\tfrac{d_{\min}(\Lambda_2)}{2} \geq \bar{r}\sqrt{N} = r\sqrt{N(2r^2N + 1)} \tag{7}$$

Using Proposition 3 with m a power of 2, one can say $r^2 = r'^2 + 2\pi \cdot \text{rad}(m)/m = r'^2 + 2\pi/N$. So $r^2N = Nr'^2 + 2\pi = Nq^2\alpha^2\sqrt{(3N/\log(3N)} + 2\pi$. Note that in R-LWE we need that $\alpha < \sqrt{\log(N)/N}$, $q = \text{poly}(N)$ and $\alpha q \geq \omega(\sqrt{\log N})$ (see Theorem 2).

When dealing with our explicit construction in paragraph IV-a, the condition on $d_{\min}$ becomes for large N:

$$\tfrac{d_{\min}(\Lambda_2)}{2} = \tfrac{\beta}{2\sqrt{2}} \geq \sqrt{r^2N(2r^2N + 1)}. \tag{8}$$

In order to satisfy condition (8) and to minimize the error probability, we choose according to (3) $\beta = q/2^{\lfloor n/2 \rfloor}$ which is equal to $q/\sqrt{N}$ if $2|n$. Hence, equation (8) becomes

$$\tfrac{q}{\sqrt{N}} \geq \sqrt{8r^2N(2r^2N + 1)} \tag{9}$$

For example, we can choose

$$\alpha = \mathcal{O}\left(\tfrac{1}{N^2(\log\log N)}\right) ; q = \mathcal{O}\left(N^2\sqrt{\log N}\,(\log\log N)^2\right).$$

It is not hard to see that these are the only values of α and q, up to logarithmic factors, that satisfy the R-LWE conditions and equation (9). With this choice, it follows from the bound (6) that the error probability can be as small as 2^{-1675} for $N = 1024$. Note that the modulus to noise ratio q/r of our scheme is of order $\tilde{O}\left(N^{7/4}\right)$, i.e. the same as in [7].

[2] More generally, to deal with the quantization error one could impose the condition that $\mathbb{P}\{\|e_Q\| < d_{\min}(\Lambda_2)/4\}$ vanishes exponentially fast.

VI. Security

We will prove that the algorithm is *IND-CPA* secure, assuming the hardness of R-DLWE$_{q,\chi}$ given two samples. This proof is generic and holds in the setting of the key generation protocol in Section IV independently of the choice of the lattices Λ_1 and Λ_2 as long as $Q_{\mathcal{P}_i(\Lambda_i)}$ can be done efficiently. We follow the same argument as Section 4.2 in [7]. We consider the adjacent games below:

Game 1	**Game 1 '**
$\mathbf{a} \xleftarrow{\$} R_q$	$\mathbf{a} \xleftarrow{\$} R_q$
$(\mathbf{b}, \mathbf{s}) \leftarrow \mathsf{Gen}(\mathbf{a})$	$(\mathbf{b}, \mathbf{s}) \leftarrow \mathsf{Gen}(\mathbf{a})$
$((\mathbf{u}, \mathbf{r}), \mathbf{k}) \leftarrow \mathsf{Encaps}(\mathbf{a}, \mathbf{b})$	$((\mathbf{u}, \mathbf{r}), \mathbf{k}) \leftarrow \mathsf{Encaps}(\mathbf{a}, \mathbf{b})$
	$\mathbf{k}^* \xleftarrow{\$} \Lambda_2/\Lambda_3$
Output $(\mathbf{a}, \mathbf{b}, (\mathbf{u}, \mathbf{r}), \mathbf{k})$	Output $(\mathbf{a}, \mathbf{b}, (\mathbf{u}, \mathbf{r}), \mathbf{k}^*)$
Game 2	**Game 3**
$\mathbf{a} \xleftarrow{\$} R_q$	$(\mathbf{a}, \mathbf{b}) \xleftarrow{\$} R_q \times R_q^\vee$
$\mathbf{b} \xleftarrow{\$} R_q^\vee$	$(\mathbf{u}, \mathbf{v}) \xleftarrow{\$} R_q^\vee \times R_q^\vee$
$((\mathbf{u}, \mathbf{r}), \mathbf{k}) \leftarrow \mathsf{Encaps}(\mathbf{a}, \mathbf{b})$	$\mathbf{r} = \text{HelpRec}(\mathbf{v})$
	$\mathbf{k}^* \xleftarrow{\$} \Lambda_2/\Lambda_3$
Output $(\mathbf{a}, \mathbf{b}, (\mathbf{u}, \mathbf{r}), \mathbf{k})$	Output $(\mathbf{a}, \mathbf{b}, (\mathbf{u}, \mathbf{r}), \mathbf{k}^*)$

Notice that Game 1 is the "real" game defined in Section II, and Game 1' is the "ideal" one. Our aim is to prove that Game 1 and Game 1' are computationally indistinguishable. We'll do so sequentially.

Clearly Game 1 and Game 2 are computationally indistinguishable under the assumption of hardness of R-DLWE$_{q,\chi}$.

To prove that Game 2 and Game 3 are computationally indistinguishable, we use the following Theorem which is essentially a consequence of the Crypto Lemma [17, Lemma 4.1.1]. It guarantees uniformity of the key without a dither.

Theorem 5. If $\mathbf{v} \in R_q^\vee$ is uniformly random, then $\mathbf{k} = \text{Rec}(\mathbf{v}, \mathbf{r})$ is uniformly random, given $\mathbf{r} = \text{HelpRec}(\mathbf{v})$.

Proof: For fixed $\mathbf{k}, \mathbf{k}' \in \Lambda_2/\Lambda_3$, we define $\forall\, \mathbf{v} \in \Lambda_0/\Lambda_3$,

$$\pi_{\mathbf{k},\mathbf{k}'}(\mathbf{v}) = (\mathbf{v} - \mathbf{k} + \mathbf{k}') \bmod \Lambda_3.$$

Notice that $\mathbf{v} \in \Lambda_0$, $(-\mathbf{k} + \mathbf{k}') \in \Lambda_2 \subseteq \Lambda_0$; then $\pi_b(\mathbf{v}) \in \Lambda_0/\Lambda_3$. Hence, $\pi_{\mathbf{k},\mathbf{k}'}$ is a permutation of $R_q^\vee$ by Lemma 1. The proof of Theorem 5 results from these lemmas:

Lemma 2. $\forall\ \mathbf{k}, \mathbf{k}' \in \Lambda_2/\Lambda_3$ and $\forall\ \mathbf{v} \in \Lambda_0/\Lambda_3$ we have $\text{HelpRec}(\mathbf{v}) = \text{HelpRec}\left(\pi_{\mathbf{k},\mathbf{k}'}(\mathbf{v})\right)$.

Proof:

$$\begin{aligned}
\mathbf{r}' &= Q_{\Lambda_1}\left(\pi_{\mathbf{k},\mathbf{k}'}(\mathbf{v})\right) \bmod \Lambda_2 \\
&= Q_{\Lambda_1}\left((\mathbf{v} - \mathbf{k} + \mathbf{k}') \bmod \Lambda_3\right) \bmod \Lambda_2 \\
&= Q_{\Lambda_1}\left(\mathbf{v} - \mathbf{k} + \mathbf{k}' - Q_{\Lambda_3}\left(\mathbf{v} - \mathbf{k} + \mathbf{k}'\right)\right) \bmod \Lambda_2 \\
&= \left(Q_{\Lambda_1}\left(\mathbf{v}\right) - \mathbf{k} + \mathbf{k}' - Q_{\Lambda_3}\left(\mathbf{v} - \mathbf{k} + \mathbf{k}'\right)\right) \bmod \Lambda_2 \\
&= Q_{\Lambda_1}\left(\mathbf{v}\right) \bmod \Lambda_2 = \mathbf{r}. \qquad \square
\end{aligned}$$

Lemma 3. Suppose that $\mathbf{k} = \text{Rec}(\mathbf{v}, \mathbf{r}) = Q_{\Lambda_2}(\mathbf{v} - \mathbf{r}) \bmod \Lambda_3$, then $\forall\, \mathbf{k}' \in \Lambda_2/\Lambda_3$ we have $\mathbf{k}' = \text{Rec}(\pi_{\mathbf{k},\mathbf{k}'}(\mathbf{v}), \mathbf{r})$.

Proof:

$$\begin{aligned}
\text{Rec}(\pi_{\mathbf{k},\mathbf{k}'}(\mathbf{v}), \mathbf{r}) &= \left[Q_{\Lambda_2}\left(\mathbf{v} - \mathbf{r}\right) - \mathbf{k} + \mathbf{k}'\right] \bmod \Lambda_3 \\
&= \left[Q_{\Lambda_2}\left(\mathbf{v} - \mathbf{r}\right) \bmod \Lambda_3 - \mathbf{k} + \mathbf{k}'\right] \bmod \Lambda_3 \\
&= \left[\mathbf{k} - \mathbf{k} + \mathbf{k}'\right] \bmod \Lambda_3 = \mathbf{k}'. \qquad \square
\end{aligned}$$

Corollary 1. $\forall\ \mathbf{k}, \mathbf{k}' \in \Lambda_2/\Lambda_3$ and $\forall\ \mathbf{v} \in \Lambda_0/\Lambda_3$, there exist $\mathbf{v}' = \pi_{\mathbf{k},\mathbf{k}'}(\mathbf{v})$ such that $\text{HelpRec}(\mathbf{v}) = \text{HelpRec}\left(\pi_{\mathbf{k},\mathbf{k}'}(\mathbf{v})\right)$, and $\mathbf{k} = \text{Rec}(\mathbf{v}, \mathbf{r}) \iff \mathbf{k}' = \text{Rec}(\pi_{\mathbf{k},\mathbf{k}'}(\mathbf{v}), \mathbf{r})$.

We conclude the proof of Theorem 5 by showing that $\mathbf{k}$ is uniform and independent of $\mathbf{r}$ when $\mathbf{v}$ is uniform:

$$\begin{aligned}
\mathbb{P}\{\mathbf{k} \mid \mathbf{r}\} &= \sum_{\mathbf{v} \in \Lambda_0/\Lambda_3} \mathbb{P}\{\mathbf{v}\} \cdot \mathbb{P}\{\mathbf{k} \mid \mathbf{r}, \mathbf{v}\} \\
&= \sum_{\mathbf{v} \in \Lambda_0/\Lambda_3} \mathbb{1}_{\left\{\substack{\mathbf{r} = \text{HelpRec}(\mathbf{v}) \\ \mathbf{k} = \text{Rec}(\mathbf{v}, \mathbf{r})}\right\}} \cdot \mathbb{P}\{\mathbf{v}\} \\
&= \sum_{\mathbf{v} \in \Lambda_0/\Lambda_3} \mathbb{1}_{\left\{\substack{\mathbf{r} = \text{HelpRec}(\pi_{\mathbf{k},\mathbf{k}'}(\mathbf{v})) \\ \mathbf{k}' = \text{Rec}(\pi_{\mathbf{k},\mathbf{k}'}(\mathbf{v}), \mathbf{r})}\right\}} \cdot \mathbb{P}\{\mathbf{v}\} \\
&= \sum_{\mathbf{v}' \in \Lambda_0/\Lambda_3} \mathbb{1}_{\left\{\substack{\mathbf{r} = \text{HelpRec}(\mathbf{v}') \\ \mathbf{k}' = \text{Rec}(\mathbf{v}', \mathbf{r})}\right\}} \cdot \mathbb{P}\{\mathbf{v}'\} \\
&= \sum_{\mathbf{v}' \in \Lambda_0/\Lambda_3} \mathbb{P}\{\mathbf{v}'\} \cdot \mathbb{P}\{\mathbf{k}' \mid \mathbf{r}, \mathbf{v}'\} = \mathbb{P}\{\mathbf{k}' \mid \mathbf{r}\}. \qquad \square
\end{aligned}$$

Returning to Game 2 and Game 3, we construct an efficient reduction $\mathcal{S}$ as follows: it takes as input two pairs $(\mathbf{a}, \mathbf{u}), (\mathbf{b}', \mathbf{v}) \in R_q \times R_q^\vee$, and outputs

$$(\mathbf{a}, \mathbf{b} = \theta(\mathbf{b}')\ ,\ (\mathbf{u}, \mathbf{r} = \text{HelpRec}(\mathbf{v})), \mathbf{k} = \text{Rec}(\mathbf{v}, \mathbf{r}))\,.$$

After that, we will take two indistinguishable inputs, and hence, by efficiency of $\mathcal{S}$, get two indistinguishable outputs. First suppose that the inputs are drawn from $A_{\mathbf{s}',\chi}$; i.e. $\mathbf{u} = \mathbf{a}\mathbf{s}' + \mathbf{e}'$ and $\mathbf{v} = \mathbf{b}'\mathbf{s}' + \mathbf{e}'' = \theta^{-1}(\mathbf{b})\mathbf{s}' + \mathbf{e}''$ for independent $\mathbf{e}', \mathbf{e}'' \longleftarrow \chi$; and then $\mathbf{a}, \mathbf{b}$ are uniformly random and independent from R_q and $R_q^\vee$ respectively (because θ is an isomorphism). Hence, the output of $\mathcal{S}$ will be exactly as in Game 2. Now suppose that the inputs given to $\mathcal{S}$ are uniformly random in $R_q \times R_q^\vee$ and independent, then the outputs of $\mathcal{S}$ are exactly as in Game 3. In fact, $\mathbf{a}, \mathbf{b}, \mathbf{u}, \mathbf{v}$ are uniform, and hence by Theorem 5, $\mathbf{k}$ is uniformly random conditioned on $\mathbf{r} = \text{HelpRec}(\mathbf{v})$.

To show that Game 3 and Game 1' are indistinguishable, we modify Game 1 and Game 2 by choosing $\mathbf{k}^* \xleftarrow{\$} \Lambda_2/\Lambda_3$ and output it instead of $\mathbf{k}$. In this case Game 1 becomes Game 1'. Let Game 2' be the modified version of Game 2. By the same reasoning as above, we can prove that Game 1' is computationally indistinguishable from Game 2' and Game 3.

Remark 3. Following the steps in [7, Section 5], we can construct a passively secure encryption scheme based on our passively secure KEM, which yields an actively secure encryption scheme and an actively secure key transport protocol.

Acknowledgments

The work of C. Saliba and L. Luzzi is supported by the INEX Paris-Seine AAP 2017. The authors would like to thank J.-P. Tillich for helpful comments.

APPENDIX
PROOF OF THEOREM 4

In the following we refer to the functions PARBW, SEQBW, RMDEC in Algorithms 1,2 and 3 of Micciancio and Nicolosi's paper [11].

A. Modification

We modify Algorithm 3 in [11] in the case $r = 0$ as follows: if $\sum_{b_j=0} \rho_j = \sum_{b_j=1} \rho_j$, then return $b_1 \cdot [1, 1, \ldots, 1]$.
It means that we choose the output vector based on the first bit of $\mathbf{b}$. Note that the decoder is still BDD with this modification.

B. Linearity of PARBW

In this subsection we will prove the following proposition:

Proposition 5. Let $\boldsymbol{\lambda} \in BW^n$ and $\mathbf{w} \in \mathbb{C}^N$ a target, where $N = 2^n$, then

$$\text{PARBW}(p, \underbrace{\boldsymbol{\lambda} + \mathbf{w}}_{\mathbf{s}}) = \boldsymbol{\lambda} + \text{PARBW}(p, \mathbf{w}), \quad \forall\, p = 4^k.$$

We will prove this by induction on p and N. The cases $p = 1$ or $N = 1$ will be proven in the next subsection. Now consider the case where $p \geq 4$ and $\mathbf{s} \notin \mathbb{C}^1$. Suppose that Proposition 5 holds for $N/2$-dimensional vectors and $p/4$ processors; we will show that it also holds for N-dimensional vectors and p processors.
Let $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_1]$ and $\mathbf{w} = [\mathbf{w}_0, \mathbf{w}_1]$. As defined in Algorithm 1, we have:

$$\begin{aligned}\mathbf{s} &:= [\mathbf{s}_0, \mathbf{s}_1] \\ &= [\boldsymbol{\lambda}_0 + \mathbf{w}_0, \boldsymbol{\lambda}_1 + \mathbf{w}_1] \\ &= [\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_1] + [\mathbf{w}_0, \mathbf{w}_1],\end{aligned}$$

and the candidate vectors are

$$\begin{bmatrix}\mathbf{z}_0(\mathbf{s}) \\ \mathbf{z}_1(\mathbf{s}) \\ \mathbf{z}_-(\mathbf{s}) \\ \mathbf{z}_+(\mathbf{s})\end{bmatrix} := \begin{bmatrix} \text{PARBW}(p/4, \boldsymbol{\lambda}_0 + \mathbf{w}_0) \\ \text{PARBW}(p/4, \boldsymbol{\lambda}_1 + \mathbf{w}_1) \\ \text{PARBW}\left(p/4, \frac{\phi}{2}[\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}_1] + \frac{\phi}{2}[\mathbf{w}_0 - \mathbf{w}_1]\right) \\ \text{PARBW}\left(p/4, \frac{\phi}{2}[\boldsymbol{\lambda}_0 + \boldsymbol{\lambda}_1] + \frac{\phi}{2}[\mathbf{w}_0 + \mathbf{w}_1]\right)\end{bmatrix}.$$

Here we use implicitly the fact that $[\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_1] \in BW^n \implies \boldsymbol{\lambda}_0, \boldsymbol{\lambda}_1 \in BW^{n-1}$. Note that

$$\begin{aligned}\mathbf{z}_0^-(\mathbf{s}) &:= [\mathbf{z}_0(\mathbf{s}), \mathbf{z}_0(\mathbf{s}) - 2\phi^{-1}\mathbf{z}_-(\mathbf{s})] \\ &= \Big[\boldsymbol{\lambda}_0 + \text{PARBW}(p/4, \mathbf{w}_0), \boldsymbol{\lambda}_0 + \text{PARBW}(p/4, \mathbf{w}_0) \\ &\quad - (\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}_1) - 2\phi^{-1}\text{PARBW}\left(p/4, \tfrac{\phi}{2}(\mathbf{w}_0 - \mathbf{w}_1)\right)\Big] \\ &= [\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_1] + \Big[\text{PARBW}(p/4, \mathbf{w}_0), \text{PARBW}(p/4, \mathbf{w}_0) \\ &\quad - 2\phi^{-1}\text{PARBW}\left(p/4, \tfrac{\phi}{2}(\mathbf{w}_0 - \mathbf{w}_1)\right)\Big] \\ &= \boldsymbol{\lambda} + [\mathbf{z}_0(\mathbf{w}), \mathbf{z}_0(\mathbf{w}) - 2\phi^{-1}\mathbf{z}_-(\mathbf{w})] \\ &= \boldsymbol{\lambda} + \mathbf{z}_0^-(\mathbf{w}).\end{aligned}$$

Following the same steps, we can prove that

$$\begin{aligned}\mathbf{z}_0^+(\mathbf{s}) &= \boldsymbol{\lambda} + \mathbf{z}_0^+(\mathbf{w}), \\ \mathbf{z}_1^-(\mathbf{s}) &= \boldsymbol{\lambda} + \mathbf{z}_1^-(\mathbf{w}), \\ \mathbf{z}_1^+(\mathbf{s}) &= \boldsymbol{\lambda} + \mathbf{z}_1^+(\mathbf{w}).\end{aligned}$$

The algorithm will return the value $\mathbf{z}'(\mathbf{s})$ in $\{\mathbf{z}_0^+(\mathbf{s}), \mathbf{z}_0^-(\mathbf{s}), \mathbf{z}_1^+(\mathbf{s}), \mathbf{z}_1^-(\mathbf{s})\}$ such that $\|\mathbf{s} - \mathbf{z}'(\mathbf{s})\|$ is minimal. This is equivalent to saying that $\mathbf{z}'(\mathbf{s}) \in \{\boldsymbol{\lambda} + \mathbf{z}_0^+(\mathbf{w}), \boldsymbol{\lambda} + \mathbf{z}_0^-(\mathbf{s}), \boldsymbol{\lambda} + \mathbf{z}_1^+(\mathbf{s}), \boldsymbol{\lambda} + \mathbf{z}_1^-(\mathbf{s})\}$ such that $\|(\boldsymbol{\lambda} + \mathbf{w}) - (\boldsymbol{\lambda} + \mathbf{z}_*^*(\mathbf{w}))\| = \|\mathbf{w} - \mathbf{z}_*^*(\mathbf{w})\|$ is minimal. In any case, the output is of the form $\boldsymbol{\lambda} + \text{PARBW}(p, \mathbf{w})$.

Our next step now is to prove that Proposition 5 holds for $p = 1$ or $\mathbf{s} \in \mathbb{C}^1$. In this case Algorithm 1 returns $\text{SEQBW}(0, \boldsymbol{\lambda} + \mathbf{w})$. This function is presented in Algorithm 2.

C. Linearity of SEQBW$(r, *)$

In this subsection we refer to the functions $t(\mathbf{s}) = (\mathbf{b}(\mathbf{s}), \rho(\mathbf{s}))$ and $\text{RMDEC}(r, t) = \text{RMDEC}(r, \mathbf{b}, \rho)$ in Algorithms 2 and 3 and to the function $\psi : \mathbb{F}_2^N \mapsto \mathbb{Z}[i]^N$ defined in [11].
Recall that each vector $\boldsymbol{\lambda}$ in BW^n can be written as

$$\boldsymbol{\lambda} = \sum_{r=0}^{n-1} \phi^r \psi(c_r) + \phi^n c_n,$$

where $c_n \in \mathbb{G}^N$ and $c_r \in RM_r^n$ for $r = 0, \ldots, n-1$.
For any $0 \leq r \leq n$, let

$$BW_r^n = \left\{ \sum_{k=r}^{n-1} \phi^{k-r}\psi(c_k) + \phi^{n-r} c_n \;:\; c_k \in RM_k^n,\, c_n \in \mathbb{G}^N \right\}$$

Proposition 6. Let $\mathbf{w} \in \mathbb{C}^N$ and $c_r \in RM_r^n$. Then

$$\tilde{\text{RMDEC}}(r, c_r \oplus \mathbf{b}(\mathbf{w}), \rho) = c_r \oplus \tilde{\text{RMDEC}}(r, \mathbf{b}(\mathbf{w}), \rho),$$

where $\tilde{\text{RMDEC}} = \text{RMDEC} \bmod 2$.

Proof: Before we start the proof, observe that for $c_r \in RM_r^n$, $\psi(c_r) \bmod 2 = c_r$.
Let's start with $n = 1$, $r = 0$. In this case it is easy to see from Algorithm 3 that ψ is the identity function and

$$\text{RMDEC}(0, c_0 \oplus \mathbf{b}(\mathbf{w}), \rho) = \psi(c_0) \oplus \text{RMDEC}(0, \mathbf{b}(\mathbf{w}), \rho).$$

So this remains true for $\tilde{\text{RMDEC}}$.
For $n = 1$ and $r = 1$ we have $2^r = N$. Thus

$$\begin{aligned}\text{RMDEC}(1, c_1 \oplus \mathbf{b}(\mathbf{w}), \rho) &= c_1 \oplus \mathbf{b}(\mathbf{w}) \\ &= c_1 \oplus \text{RMDEC}(1, \mathbf{b}(\mathbf{w}), \rho)\end{aligned}$$

This remains also true for $\tilde{\text{RMDEC}}$. Now we continue by induction: suppose Proposition 6 holds for $(r-1, n-1)$ and $(r, n-1)$. We want to show that it holds for (r, n). Following the notation in Algorithm 3, we find:

$[t^0, t^1] \longleftarrow t = (c_r \oplus \mathbf{b}(\mathbf{w}), \rho)$,
$c_r \in RM_r^n \Rightarrow c_r = [u', u' \oplus v'],\ u' \in RM_r^{n-1}, v' \in RM_{r-1}^{n-1}$,
$t_j^+ = \left(v'_j \oplus b(\mathbf{w})_j^+, \min(\rho_j^0, \rho_j^1)\right)$.

Note that

$$\begin{aligned}v \bmod 2 &= \tilde{\text{RMDEC}}(r-1, t^+) \\ &= \tilde{\text{RMDEC}}(r-1, v' \oplus \mathbf{b}^+(\mathbf{w}), \rho^+) \\ &= v' \oplus \tilde{\text{RMDEC}}(r-1, \mathbf{b}^+(\mathbf{w}), \rho^+) \\ &= v' \oplus \tilde{\text{RMDEC}}(r-1, t(\mathbf{w})^+).\end{aligned}$$

Now we compute the vector u. If $v_j = v'_j \oplus b_j^+(\mathbf{w}) \bmod 2$, then

$$t_j^- = \left(u'_j \oplus b_j^0(\mathbf{w}), (\rho_j^0 + \rho_j^1)/2\right).$$

Otherwise,

$$t_j^- = \left(u'_j \oplus b_j^0(\mathbf{w}) \oplus \textsc{Eval}(\rho_j^0 < \rho_j^1), |\rho_j^0 - \rho_j^1|/2\right).$$

Then we have

$$\begin{aligned} u \bmod 2 &= \tilde{\text{RM}}\textsc{Dec}(r, t^-) \\ &= \tilde{\text{RM}}\textsc{Dec}(r, u' \oplus \mathbf{b}^-(\mathbf{w}), \rho^-) \\ &= u' \oplus \tilde{\text{RM}}\textsc{Dec}(r, \mathbf{b}^-(\mathbf{w}), \rho^-) \\ &= u' \oplus \tilde{\text{RM}}\textsc{Dec}(r, t(\mathbf{w})^-). \end{aligned}$$

Hence we obtain:

$$\begin{aligned} &\tilde{\text{RM}}\textsc{Dec}(r, c_r \oplus \mathbf{b}(\mathbf{w}), \rho) \\ &= [u \bmod 2, (u+v) \bmod 2] \\ &= [u' \oplus \tilde{\text{RM}}\textsc{Dec}(r, t(\mathbf{w})^-), u' \oplus \\ &\quad \tilde{\text{RM}}\textsc{Dec}(r, t(\mathbf{w})^-) \oplus v' \oplus \tilde{\text{RM}}\textsc{Dec}(r-1, t(\mathbf{w})^+)] \\ &= [u', u' \oplus v'] \oplus [\tilde{\text{RM}}\textsc{Dec}(r, t(\mathbf{w})^-), \\ &\quad \tilde{\text{RM}}\textsc{Dec}(r, t(\mathbf{w})^-) \oplus \tilde{\text{RM}}\textsc{Dec}(r-1, t(\mathbf{w})^+)] \\ &= c_r \oplus \tilde{\text{RM}}\textsc{Dec}(r, \mathbf{b}(\mathbf{w}), \rho). \end{aligned}$$

□

Lemma 4. If $\mathbf{d} \in BW_r^n$, then $(\mathbf{d}, \mathbf{d}) \in BW_r^{n+1}$.

Proof: Suppose $\mathbf{d} = \psi(c_r) + \phi \cdot \psi(c_{r+1}) + \cdots + \phi^{n-1-r} \cdot \psi(c_{n-1}) + \phi^{n-r} \cdot c_n$, where $c_i \in RM_i^n$.
Hence,

$$\begin{aligned} (\mathbf{d}, \mathbf{d}) &= \left(\psi(c_r) + \cdots + \phi^{n-r} \cdot c_n, \psi(c_r) + \cdots + \phi^{n-r} \cdot c_n\right) \\ &= (\psi(c_r), \psi(c_r)) + \cdots + \phi^{n-r} \cdot (c_n, c_n) \\ &= \psi(c_r, c_r) + \cdots + \phi^{n-r} \cdot (c_n, c_n), \end{aligned}$$

noting that $(\psi(c_i), \psi(c_i)) = \psi(c_i, c_i)$. Also note that if $c_i \in RM_i^n$, then $(c_i, c_i) \in RM_i^{n+1}$. This proves that $(\mathbf{d}, \mathbf{d}) \in BW_r^{n+1}$. □

Lemma 5. If $\mathbf{d}' \in BW_{r-1}^n$, then $(\mathbf{0}, \mathbf{d}') \in BW_r^{n+1}$.

Proof: Let $\mathbf{d}' = \psi(c'_{r-1}) + \phi \cdot \psi(c'_r) + \cdots + \phi^{n-r} \cdot \psi(c'_{n-1}) + \phi^{n-r+1} \cdot c'_n$, where $c'_i \in RM_i^n$.

$$\begin{aligned} (\mathbf{0}, \mathbf{d}') &= (\psi(0), \psi(0) + \mathbf{d}') \\ &= (\psi(0), \psi(0) + \psi(c'_{r-1})) + \cdots + \phi^{n-r+1} \cdot (0, c'_n) \\ &= \psi(\underbrace{0, 0 \oplus c'_{r-1}}_{\in RM_r^{n+1}}) + \cdots + \phi^{n-r+1} \cdot \underbrace{(0, 0 + c'_n)}_{\in \mathbb{Z}[i]^{2^{n+1}}} \end{aligned}$$

This shows that $(\mathbf{0}, \mathbf{d}') \in BW_r^{n+1}$. □

Lemma 6. For any $a, b \in RM_r^n$ and $r < n$ we have $\psi(a \oplus b) = \psi(a) + \psi(b) + 2\mathbf{d}$ for some $\mathbf{d} \in BW_{r+1}^n$.

Proof: Let's first define the Schur product: for binary vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^N$, we set

$$\mathbf{x} * \mathbf{y} = (x_1 y_1, \ldots, x_N y_N).$$

For $n = 1$, and $r = 0$, we have for $a, b \in RM_0^1 = \{(0,0), (1,1)\}$:

$$\psi(a \oplus b) = a \oplus b = a + b - 2(a * b).$$

So here $\mathbf{d} = -(a * b) \in \mathbb{Z}[i]^2 = BW_1^1$.
Suppose that the hypothesis holds for n, for all $r < n$. Let $(a, b) \in RM_r^{n+1}$.
- If $r < n$, then write $a = (u, u \oplus v)$ and $b = (u', u' \oplus v')$, where $u, u' \in RM_r^n$ and $v, v' \in RM_{r-1}^n$. Then

$$\begin{aligned} \psi(a \oplus b) &= \psi(u \oplus u', u \oplus u' \oplus v \oplus v') \\ &= (\psi(u \oplus u'), \psi(u \oplus u') + \psi(v \oplus v')) \end{aligned}$$

By inductive hypothesis

$$\begin{cases} \psi(u \oplus u') = \psi(u) + \psi(u') + 2\mathbf{d};\ \mathbf{d} \in BW_{r+1}^n \\ \psi(v \oplus v') = \psi(v) + \psi(v') + 2\mathbf{d}';\ \mathbf{d}' \in BW_r^n \end{cases}$$

So we can write

$$\begin{aligned} \psi(a \oplus b) &= (\psi(u) + \psi(u') + 2\mathbf{d}, \\ &\qquad \psi(u) + \psi(u') + 2\mathbf{d} + \psi(v) + \psi(v') + 2\mathbf{d}') \\ &= (\psi(u), \psi(u) + \psi(v)) \\ &\quad + (\psi(u'), \psi(u') + \psi(v')) + 2(\mathbf{d}, \mathbf{d} + \mathbf{d}') \\ &= \psi(a) + \psi(b) + 2(\mathbf{d}, \mathbf{d} + \mathbf{d}') \end{aligned}$$

But $(\mathbf{d}, \mathbf{d}) \in BW_{r+1}^{n+1}$ and $(0, \mathbf{d}') \in BW_{r+1}^{n+1}$, so $(\mathbf{d}, \mathbf{d} + \mathbf{d}') \in BW_{r+1}^{n+1}$.
- If $r = n$, $BW_{n+1}^{n+1} = \mathbb{Z}[i]^{2N}$, so the statement is trivially true. □

Proposition 7. For $\boldsymbol{\lambda}_r \in BW_r^n$ and $\mathbf{w} \in \mathbb{C}^N$, we have

$$\textsc{SeqBW}(r, \boldsymbol{\lambda}_r + \mathbf{w}) = \boldsymbol{\lambda}_r + \textsc{SeqBW}(r, \mathbf{w}).$$

Proof: We will proceed by decreasing induction on r.
If $r \geq n$, then SeqBW is nothing more than the rounding function, and the property holds.
Now suppose that the hypothesis remains true for $r+1$; let's prove it for r:

$$\begin{aligned} &\textsc{SeqBW}(r, \boldsymbol{\lambda}_r + \mathbf{w}) \\ &= \text{RM}\textsc{Dec}(r, t(\boldsymbol{\lambda}_r + \mathbf{w}))) \\ &+ \phi\, \textsc{SeqBW}\left(r+1, \frac{\boldsymbol{\lambda}_r + \mathbf{w} - \text{RM}\textsc{Dec}(r, t(\boldsymbol{\lambda}_r + \mathbf{w}))}{\phi}\right) \\ &= \psi\left(\tilde{\text{RM}}\textsc{Dec}(r, t(\boldsymbol{\lambda}_r + \mathbf{w}))\right) \\ &+ \phi\, \textsc{SeqBW}\left(r+1, \frac{\boldsymbol{\lambda}_r + \mathbf{w} - \psi\left(\tilde{\text{RM}}\textsc{Dec}(r, t(\boldsymbol{\lambda}_r + \mathbf{w}))\right)}{\phi}\right) \\ &\overset{(a)}{=} \psi\left(\tilde{\text{RM}}\textsc{Dec}(r, c_r \oplus \mathbf{b}(\mathbf{w}), \rho)\right) \\ &+ \phi\, \textsc{SeqBW}\left(r+1, \boldsymbol{\lambda}_{r+1} + \frac{\psi(c_r) + \mathbf{w} - \psi\left(\tilde{\text{RM}}\textsc{Dec}(r, c_r \oplus \mathbf{b}(\mathbf{w}), \rho)\right)}{\phi}\right) \\ &= \psi\left(c_r \oplus \tilde{\text{RM}}\textsc{Dec}(r, \mathbf{b}(\mathbf{w}), \rho)\right) + \phi \cdot \boldsymbol{\lambda}_{r+1} \\ &+ \phi\, \textsc{SeqBW}\left(r+1, \frac{\psi(c_r) + \mathbf{w} - \psi\left(c_r \oplus \tilde{\text{RM}}\textsc{Dec}(r, \mathbf{b}(\mathbf{w}), \rho)\right)}{\phi}\right) \\ &\overset{(b)}{=} \psi(c_r) + \psi\left(\tilde{\text{RM}}\textsc{Dec}(r, \mathbf{b}(\mathbf{w}), \rho)\right) + 2\mathbf{d} + \phi \cdot \boldsymbol{\lambda}_{r+1} \\ &+ \phi\, \textsc{SeqBW}\left(r+1, \frac{\mathbf{w} - \psi\left(\tilde{\text{RM}}\textsc{Dec}(r, \mathbf{b}(\mathbf{w}), \rho)\right) - 2\mathbf{d}}{\phi}\right), \end{aligned}$$

where (a) follows from the fact that $\boldsymbol{\lambda}_r = \psi(c_r) + \phi \cdot \boldsymbol{\lambda}_{r+1}$ for $c_r \in RM_r^n$ and $\boldsymbol{\lambda}_{r+1} \in BW_{r+1}^n$, and (b) follows from Lemma 6. Observe that $\frac{2\mathbf{d}}{\phi} = (1-i)\mathbf{d} \in BW_{r+1}^n$. By inductive hypothesis for $\text{SEQBW}(r+1, *)$:

$$
\begin{aligned}
&\text{SEQBW}(r, \boldsymbol{\lambda}_r + \mathbf{w}) \\
&= \boldsymbol{\lambda}_r + \psi\left(\tilde{\text{RMDEC}}(r, \mathbf{b}(\mathbf{w}), \rho)\right) + 2\mathbf{d} - 2\mathbf{d} + \\
&\quad \phi \cdot \text{SEQBW}\left(r+1, \frac{\mathbf{w} - \psi\left(\tilde{\text{RMDEC}}(r, \mathbf{b}(\mathbf{w}), \rho)\right)}{\phi}\right) \\
&= \boldsymbol{\lambda}_r + \text{RMDEC}(r, \mathbf{b}(\mathbf{w}), \rho) + \\
&\quad \phi \cdot \text{SEQBW}\left(r+1, \frac{\mathbf{w} - \text{RMDEC}(r, \mathbf{b}(\mathbf{w}), \rho)}{\phi}\right) \\
&= \boldsymbol{\lambda}_r + \text{SEQBW}(r, \mathbf{w}).
\end{aligned}
$$

□

REFERENCES

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[2] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[3] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM (JACM)*, vol. 56, no. 6, p. 34, 2009.

[4] C. Peikert and B. Waters, "Lossy trapdoor functions and their applications," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1803–1844, 2011.

[5] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2010, pp. 1–23.

[6] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 3, p. 13, 2014.

[7] C. Peikert, "Lattice cryptography for the internet," in *International Workshop on Post-Quantum Cryptography*. Springer, 2014, pp. 197–219.

[8] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key exchange-a new hope." in *USENIX Security Symposium*, vol. 2016, 2016.

[9] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "NewHope without reconciliation." *IACR Cryptology ePrint Archive*, vol. 2016, p. 1157, 2016.

[10] J. H. Conway and N. J. A. Sloane, *Sphere packings, lattices and groups*. Springer Science & Business Media, 2013, vol. 290.

[11] D. Micciancio and A. Nicolosi, "Efficient bounded distance decoders for Barnes-Wall lattices," in *2008 IEEE International Symposium on Information Theory*. IEEE, 2008, pp. 2484–2488.

[12] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," in *Annual International Cryptology Conference*. Springer, 1999, pp. 537–554.

[13] J.-P. D'Anvers, Q. Guo, T. Johansson, A. Nilsson, F. Vercauteren, and I. Verbauwhede, "Decryption failure attacks on IND-CCA secure lattice-based schemes," in *IACR International Workshop on Public Key Cryptography*. Springer, 2019, pp. 565–598.

[14] F. Oggier and J.-C. Belfiore, "Enabling multiplication in lattice codes via construction A," *IEEE Information Theory Workshop*, 2013.

[15] D. Hsu, S. Kakade, and T. Zhang, "A tail inequality for quadratic forms of subgaussian random vectors," *Electronic Communications in Probability*, vol. 17, 2012.

[16] V. Lyubashevsky, C. Peikert, and O. Regev, "A toolkit for ring-LWE cryptography," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2013, pp. 35–54.

[17] R. Zamir, *Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation, and Multiuser Information Theory*. Cambridge University Press, 2014.