The Importance of the Public Global Parameter On Ring-LWE problem-based Key Encapsulation Mechanisms

Reynaldo C. Villena & Routo Terada
reynaldo@ime.usp.br rt@ime.usp.br

Institute of Mathematics and Statistics
University of São Paulo
SP – Brazil

September, 2022

Ring-LWE Problem

Ring-LWE

The Ring-LWE problem:

- Is assumed as hard [Regev 2009, Peikert 2009].
- Is promising, due to the provable security and high efficiency [Lindner and Peikert 2011, Regev 2009].

Ring-LWE

The Ring-LWE problem:

- Is assumed as hard [Regev 2009, Peikert 2009].
- Is promising, due to the provable security and high efficiency [Lindner and Peikert 2011, Regev 2009].

The Ring-LWE problem is used as the basis of:

- Public key encryption schemes [de Clercq et al. 2015, Lyubashevsky et al. 2013],
- Digital signatures [Barreto et al. 2016, Wu et al. 2012]
- Key Encapsulation Mechanisms (KEM) [Bos et al. 2015, Alkim et al. 2017],
- Homomorphic encryptions [Fan and Vercauteren 2012, Roy et al. 2016].

Ring-LWE problem

The Ring-LWE problem fixes a power of two n and modulus q.

Let $\mathbb{Z}_q$ be the residue class ring modulo q

Let $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$ denote the polynomial ring modulo $x^n + 1$ where the coefficients are in $\mathbb{Z}_q$.

Ring-LWE problem

The Ring-LWE problem fixes a power of two n and modulus q.

Let $\mathbb{Z}_q$ be the residue class ring modulo q

Let $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$ denote the polynomial ring modulo $x^n + 1$ where the coefficients are in $\mathbb{Z}_q$.

For $\mathbf{s} \in \mathcal{R}_q$ called as secret, the Ring-LWE distribution $A_{\mathbf{s},\chi}$ over $\mathcal{R}_q \times \mathcal{R}_q$ is sampled by choosing $\mathbf{a} \in \mathcal{R}_q$ uniformly at random, choosing $\mathbf{e} \rightarrow \chi_\sigma^n$, and outputting $(\mathbf{a}, \mathbf{a}.\mathbf{s} + \mathbf{e})$.

Ring-LWE problem

The Ring-LWE problem fixes a power of two n and modulus q.

Let $\mathbb{Z}_q$ be the residue class ring modulo q

Let $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$ denote the polynomial ring modulo $x^n + 1$ where the coefficients are in $\mathbb{Z}_q$.

For $\mathbf{s} \in \mathcal{R}_q$ called as secret, the Ring-LWE distribution $A_{\mathbf{s},\chi}$ over $\mathcal{R}_q \times \mathcal{R}_q$ is sampled by choosing $\mathbf{a} \in \mathcal{R}_q$ uniformly at random, choosing $\mathbf{e} \to \chi_\sigma^n$, and outputting $(\mathbf{a}, \mathbf{a}.\mathbf{s} + \mathbf{e})$.

Search Ring-LWE$_{q,\chi,k}$:

Given k independent samples $(\mathbf{a}_i, \mathbf{b}_i) \in \mathcal{R}_q \times \mathcal{R}_q$ drawn from $A_{s,\chi}$ for a uniformly random $\mathbf{s} \in \mathcal{R}_q$ (fixed for all samples), find $\mathbf{s}$.

Key Encapsulation Mechanisms based on Ring-LWE

Ring-LWE KEM

Common parameter: $\mathbf{a} \xleftarrow{\$} \mathcal{R}_q$		
Alice		**Bob**
$\mathbf{s}_A, \mathbf{e}_A \xleftarrow{\$} \chi_\sigma^n$		
$\mathbf{p}_A \leftarrow \mathbf{a}.\mathbf{s}_A + \mathbf{e}_A$	$\xrightarrow{\mathbf{p}_A}$	$\mathbf{s}_B, \mathbf{e}_B, \mathbf{e}'_B \xleftarrow{\$} \chi_\sigma^n$
		$\mathbf{p}_B \leftarrow \mathbf{a}.\mathbf{s}_B + \mathbf{e}_B$
		$\mathbf{s}_{K_B} \xleftarrow{\$} \{0,1\}^n$
		$\mathbf{k} \leftarrow \left\lfloor \frac{q}{2} \right\rfloor \mathbf{s}_{K_B}$
$\mathbf{c}' \leftarrow \mathbf{c} - \mathbf{p}_B.\mathbf{s}_A$	$\xleftarrow{(\mathbf{p}_B, \mathbf{c})}$	$\mathbf{c} \leftarrow \mathbf{p}_A.\mathbf{s}_B + \mathbf{e}'_B + \mathbf{k}$
$\mathbf{s}_{K_A} \leftarrow \left\lceil \mathbf{c}'.\frac{2}{q} \right\rfloor \mod 2$		

Figure: Ring-LWE KEM

NewHope KEM

Common parameter: $\mathbf{a} \xleftarrow{\$} \mathcal{R}_q$

Alice		**Bob**
$\mathbf{s}_A, \mathbf{e}_A \xleftarrow{\$} \psi_8^n$		
$\mathbf{p}_A \leftarrow \mathbf{a}.\mathbf{s}_A + \mathbf{e}_A$	$\xrightarrow{\mathbf{p}_A}$	$\mathbf{s}_B, \mathbf{e}_B, \mathbf{e}'_B \xleftarrow{\$} \psi_8^n$
		$\mathbf{p}_B \leftarrow \mathbf{a}.\mathbf{s}_B + \mathbf{e}_B$
		$\mathbf{v}_B \xleftarrow{\$} \{0,1\}^{256}$
		$\mathbf{v}'_B \leftarrow \text{SHA3-256}(\mathbf{v}_B)$
		$\mathbf{k} \leftarrow \textbf{Encode}(\mathbf{v}'_B)$
		$\mathbf{c} \leftarrow \mathbf{p}_A.\mathbf{s}_B + \mathbf{e}'_B + \mathbf{k}$
$\mathbf{c}' \leftarrow \textbf{Decompress}(\overline{\mathbf{c}})$	$\xleftarrow{(\mathbf{p}_B, \overline{\mathbf{c}})}$	$\overline{\mathbf{c}} \leftarrow \textbf{Compress}(\mathbf{c})$
$\mathbf{k}' \leftarrow \mathbf{c}' - \mathbf{p}_B.\mathbf{s}_A$		$\mathbf{s}_{K_B} \leftarrow \text{SHA3-256}(\mathbf{v}'_B)$
$\mathbf{v}'_A \leftarrow \textbf{Decode}(\mathbf{k}')$		
$\mathbf{s}_{K_A} \leftarrow \text{SHA3-256}(\mathbf{v}'_A)$		

Figure: NewHope KEM

Bad Values for the Public Global Parameter

Case 1 (Trivial Case)

Let $\mathbf{a}$ be an integer ($\mathbf{a} = m$)

Case 1 (Trivial Case)

Let $\mathbf{a}$ be an integer ($\mathbf{a} = m$)
Alice generates her public key $\mathbf{p}_A = \mathbf{s}_A.\mathbf{a} + \mathbf{e}_A$.

$$\mathbf{p}_A[i] = \mathbf{s}_A[i].m + \mathbf{e}_A[i] \qquad \text{for } 0 \leq i \leq n-1$$

Case 1 (Trivial Case)

Let $\mathbf{a}$ be an integer ($\mathbf{a} = m$)
Alice generates her public key $\mathbf{p}_A = \mathbf{s}_A.\mathbf{a} + \mathbf{e}_A$.

$$\mathbf{p}_A[i] = \mathbf{s}_A[i].m + \mathbf{e}_A[i] \qquad \text{for } 0 \leq i \leq n-1$$

We can recover some coefficients of $\mathbf{s}_A$ applying $\lceil . \rfloor$ on $\frac{\mathbf{p}_A}{m}$.

Case 1 (Trivial Case)

Let $\mathbf{a}$ be an integer ($\mathbf{a} = m$)
Alice generates her public key $\mathbf{p}_A = \mathbf{s}_A.\mathbf{a} + \mathbf{e}_A$.

$$\mathbf{p}_A[i] = \mathbf{s}_A[i].m + \mathbf{e}_A[i] \qquad \text{for } 0 \leq i \leq n-1$$

We can recover some coefficients of $\mathbf{s}_A$ applying $\lceil . \rfloor$ on $\frac{\mathbf{p}_A}{m}$.

$$\left\lceil \frac{\mathbf{p}_A[i]}{m} \right\rfloor = \left\lceil \mathbf{s}_A[i] + \frac{\mathbf{e}_A[i]}{m} \right\rfloor = \mathbf{s}_A[i] + \left\lceil \frac{\mathbf{e}_A[i]}{m} \right\rfloor$$

Where the i-th coefficient of $\mathbf{s}_A$ can be retrieved with no error if $-\frac{1}{2} < \frac{\mathbf{e}_A[i]}{m} < \frac{1}{2}$.

Case 1 (Trivial Case)

Let $\mathbf{a}$ be an integer ($\mathbf{a} = m$)
Alice generates her public key $\mathbf{p}_A = \mathbf{s}_A.\mathbf{a} + \mathbf{e}_A$.

$$\mathbf{p}_A[i] = \mathbf{s}_A[i].m + \mathbf{e}_A[i] \qquad \text{for } 0 \leq i \leq n-1$$

We can recover some coefficients of $\mathbf{s}_A$ applying $\lceil . \rfloor$ on $\frac{\mathbf{p}_A}{m}$.

$$\left\lceil \frac{\mathbf{p}_A[i]}{m} \right\rfloor = \left\lceil \mathbf{s}_A[i] + \frac{\mathbf{e}_A[i]}{m} \right\rfloor = \mathbf{s}_A[i] + \left\lceil \frac{\mathbf{e}_A[i]}{m} \right\rfloor$$

Where the i-th coefficient of $\mathbf{s}_A$ can be retrieved with no error if $-\frac{1}{2} < \frac{\mathbf{e}_A[i]}{m} < \frac{1}{2}$.

On NewHope:

The value $\mathbf{e}_A \in \psi_8^n$ ($-8 \leq \mathbf{e}_A[i] \leq 8$), therefore $m \geq 17$ because $-\frac{1}{2} < \frac{\mathbf{e}_A[i]}{m} < \frac{1}{2}$.

Case 1 (Trivial Case)

Let $\mathbf{a}$ be an integer ($\mathbf{a} = m$)
Alice generates her public key $\mathbf{p}_A = \mathbf{s}_A.\mathbf{a} + \mathbf{e}_A$.

$$\mathbf{p}_A[i] = \mathbf{s}_A[i].m + \mathbf{e}_A[i] \qquad \text{for } 0 \leq i \leq n-1$$

We can recover some coefficients of $\mathbf{s}_A$ applying $\lceil . \rfloor$ on $\frac{\mathbf{p}_A}{m}$.

$$\left\lceil \frac{\mathbf{p}_A[i]}{m} \right\rfloor = \left\lceil \mathbf{s}_A[i] + \frac{\mathbf{e}_A[i]}{m} \right\rfloor = \mathbf{s}_A[i] + \left\lceil \frac{\mathbf{e}_A[i]}{m} \right\rfloor$$

Where the i-th coefficient of $\mathbf{s}_A$ can be retrieved with no error if $-\frac{1}{2} < \frac{\mathbf{e}_A[i]}{m} < \frac{1}{2}$.

On NewHope:

The value $\mathbf{e}_A \in \psi_8^n$ ($-8 \leq \mathbf{e}_A[i] \leq 8$), therefore $m \geq 17$ because $-\frac{1}{2} < \frac{\mathbf{e}_A[i]}{m} < \frac{1}{2}$.

Note:

The value of $\mathbf{a}$ being an integer would be suspicious for the participants. Alice and Bob can deny to share a secret using this suspect value of $\mathbf{a}$.

Case 2

Let $\mathbf{c}$ be a polynomial ($\mathbf{c} \in \mathcal{R}_q$) such that $\mathbf{a}.\mathbf{c} = m$

Case 2

Let $\mathbf{c}$ be a polynomial ($\mathbf{c} \in \mathcal{R}_q$) such that $\mathbf{a}.\mathbf{c} = m$

Alice generates her public key $\mathbf{p}_A = \mathbf{s}_A.\mathbf{a} + \mathbf{e}_A$.

$$\mathbf{p}_A.\mathbf{c} = \mathbf{s}_A.\mathbf{a}.\mathbf{c} + \mathbf{e}_A.\mathbf{c} = \mathbf{s}_A.m + \mathbf{e}_A.\mathbf{c} \qquad \text{because } \mathbf{a}.\mathbf{c} = m$$

Case 2

Let $\mathbf{c}$ be a polynomial ($\mathbf{c} \in \mathcal{R}_q$) such that $\mathbf{a}.\mathbf{c} = m$

Alice generates her public key $\mathbf{p}_A = \mathbf{s}_A.\mathbf{a} + \mathbf{e}_A$.

$$\mathbf{p}_A.\mathbf{c} = \mathbf{s}_A.\mathbf{a}.\mathbf{c} + \mathbf{e}_A.\mathbf{c} = \mathbf{s}_A.m + \mathbf{e}_A.\mathbf{c} \qquad \text{because } \mathbf{a}.\mathbf{c} = m$$

We can recover some coefficients of $\mathbf{s}_A$ applying $\lceil . \rfloor$ on $\frac{\mathbf{p}_A.\mathbf{c}}{m}$.

$$\left\lceil \frac{(\mathbf{p}_A.\mathbf{c})[i]}{m} \right\rfloor = \mathbf{s}_A[i] + \left\lceil \frac{(\mathbf{e}_A.\mathbf{c})[i]}{m} \right\rfloor \qquad \text{for } 0 \leq i \leq n-1$$

Where the i-th coefficient of $\mathbf{s}_A$ can be retrieved with no error if $-\frac{1}{2} < \frac{(\mathbf{e}_A.\mathbf{c})[i]}{m} < \frac{1}{2}$.

Case 2

Let $\mathbf{c}$ be a polynomial ($\mathbf{c} \in \mathcal{R}_q$) such that $\mathbf{a}.\mathbf{c} = m$

Alice generates her public key $\mathbf{p}_A = \mathbf{s}_A.\mathbf{a} + \mathbf{e}_A$.

$$\mathbf{p}_A.\mathbf{c} = \mathbf{s}_A.\mathbf{a}.\mathbf{c} + \mathbf{e}_A.\mathbf{c} = \mathbf{s}_A.m + \mathbf{e}_A.\mathbf{c} \qquad \text{because } \mathbf{a}.\mathbf{c} = m$$

We can recover some coefficients of $\mathbf{s}_A$ applying $\lceil . \rfloor$ on $\frac{\mathbf{p}_A.\mathbf{c}}{m}$.

$$\left\lceil \frac{(\mathbf{p}_A.\mathbf{c})[i]}{m} \right\rfloor = \mathbf{s}_A[i] + \left\lceil \frac{(\mathbf{e}_A.\mathbf{c})[i]}{m} \right\rfloor \qquad \text{for } 0 \leq i \leq n-1$$

Where the i-th coefficient of $\mathbf{s}_A$ can be retrieved with no error if $-\frac{1}{2} < \frac{(\mathbf{e}_A.\mathbf{c})[i]}{m} < \frac{1}{2}$. Therefore the polynomial $\mathbf{e}_A.\mathbf{c}$ should be small.

On NewHope:

- The value $\mathbf{e}_A \in \psi_8^n$ and $\mathbf{e}_A.\mathbf{c}$ should be small therefore $\mathbf{c}$ should belong to ψ_μ^n where μ is a small integer.
- The parameter $\mathbf{a}$ that leaks information about secret keys can be generated using the formula $\mathbf{a} = m(\psi_\mu^n)^{-1}$.

Case 2

Note:

The parameter $\mathbf{a} = m(\psi_\mu^n)^{-1}$.

- The value of $\mathbf{a}$ is a polynomial with different coefficients in $\mathbb{Z}_q$, being less suspicious for the participants.
- Both participants can calculate $\mathbf{a}^{-1}m$ (brute force to determine the value of m). If the result $\mathbf{c} = \mathbf{a}^{-1}m$ is a small polynomial ($\mathbf{c} \in \psi_\mu^n$) then it is possible that $\mathbf{a}$ leaks information about the secret key.

Case 3 (Adding error to Case 2)

Let $\mathbf{c}$ be a polynomial ($\mathbf{c} \in \mathcal{R}_q$) such that $\mathbf{a}.\mathbf{c} = m + \psi_\nu^n$

Case 3 (Adding error to Case 2)

Let $\mathbf{c}$ be a polynomial ($\mathbf{c} \in \mathcal{R}_q$) such that $\mathbf{a}.\mathbf{c} = m + \psi_\nu^n$

Alice generates her public key $\mathbf{p}_A = \mathbf{s}_A.\mathbf{a} + \mathbf{e}_A$.

$$\begin{aligned} \mathbf{p}_A.\mathbf{c} = \mathbf{s}_A.\mathbf{a}.\mathbf{c} + \mathbf{e}_A.\mathbf{c} &= \mathbf{s}_A.(m + \psi_\nu^n) + \mathbf{e}_A.\mathbf{c} \qquad \text{because } \mathbf{a}.\mathbf{c} = m + \psi_\nu^n \\ &= \mathbf{s}_A.m + \mathbf{s}_A.\psi_\nu^n + \mathbf{e}_A.\mathbf{c} \end{aligned}$$

Case 3 (Adding error to Case 2)

Let $\mathbf{c}$ be a polynomial ($\mathbf{c} \in \mathcal{R}_q$) such that $\mathbf{a}.\mathbf{c} = m + \psi_\nu^n$

Alice generates her public key $\mathbf{p}_A = \mathbf{s}_A.\mathbf{a} + \mathbf{e}_A$.

$$\begin{aligned} \mathbf{p}_A.\mathbf{c} = \mathbf{s}_A.\mathbf{a}.\mathbf{c} + \mathbf{e}_A.\mathbf{c} &= \mathbf{s}_A.(m + \psi_\nu^n) + \mathbf{e}_A.\mathbf{c} \qquad \text{because } \mathbf{a}.\mathbf{c} = m + \psi_\nu^n \\ &= \mathbf{s}_A.m + \mathbf{s}_A.\psi_\nu^n + \mathbf{e}_A.\mathbf{c} \end{aligned}$$

We can recover some coefficients of $\mathbf{s}_A$ applying $\lceil . \rfloor$ on $\frac{\mathbf{p}_A.\mathbf{c}}{m}$.

$$\left\lceil \frac{(\mathbf{p}_A.\mathbf{c})[i]}{m} \right\rfloor = \mathbf{s}_A[i] + \left\lceil \frac{(\mathbf{s}_A.\psi_\nu^n + \mathbf{e}_A.\mathbf{c})[i]}{m} \right\rfloor \qquad \text{for } 0 \le i \le n-1$$

The i-th coefficient of $\mathbf{s}_A$ can be retrieved with no error if $-\frac{1}{2} < \frac{(\mathbf{s}_A.\psi_\nu^n + \mathbf{e}_A.\mathbf{c})[i]}{m} < \frac{1}{2}$
Therefore the polynomial $\mathbf{s}_A.\psi_\nu^n + \mathbf{e}_A.\mathbf{c}$ should be small.

Case 3 (Adding error to Case 2)

Let $\mathbf{c}$ be a polynomial ($\mathbf{c} \in \mathcal{R}_q$) such that $\mathbf{a}.\mathbf{c} = m + \psi_\nu^n$

Alice generates her public key $\mathbf{p}_A = \mathbf{s}_A.\mathbf{a} + \mathbf{e}_A$.

$$\begin{aligned} \mathbf{p}_A.\mathbf{c} = \mathbf{s}_A.\mathbf{a}.\mathbf{c} + \mathbf{e}_A.\mathbf{c} &= \mathbf{s}_A.(m + \psi_\nu^n) + \mathbf{e}_A.\mathbf{c} \qquad \text{because } \mathbf{a}.\mathbf{c} = m + \psi_\nu^n \\ &= \mathbf{s}_A.m + \mathbf{s}_A.\psi_\nu^n + \mathbf{e}_A.\mathbf{c} \end{aligned}$$

We can recover some coefficients of $\mathbf{s}_A$ applying $\lceil . \rfloor$ on $\frac{\mathbf{p}_A.\mathbf{c}}{m}$.

$$\left\lceil \frac{(\mathbf{p}_A.\mathbf{c})[i]}{m} \right\rfloor = \mathbf{s}_A[i] + \left\lceil \frac{(\mathbf{s}_A.\psi_\nu^n + \mathbf{e}_A.\mathbf{c})[i]}{m} \right\rfloor \qquad \text{for } 0 \leq i \leq n-1$$

The i-th coefficient of $\mathbf{s}_A$ can be retrieved with no error if $-\frac{1}{2} < \frac{(\mathbf{s}_A.\psi_\nu^n + \mathbf{e}_A.\mathbf{c})[i]}{m} < \frac{1}{2}$
Therefore the polynomial $\mathbf{s}_A.\psi_\nu^n + \mathbf{e}_A.\mathbf{c}$ should be small.

On NewHope:

- The values $\mathbf{s}_A, \mathbf{e}_A \in \psi_8^n$ and $\mathbf{s}_A.\psi_\nu^n + \mathbf{e}_A.\mathbf{c}$ should be small therefore $\mathbf{c}$ should belong to ψ_μ^n where μ is a small integer.
- The parameter $\mathbf{a}$ that leaks information about secret keys can be generated using the formula $\mathbf{a} = (m + \psi_\nu^n)\mathbf{c}^{-1}$.

Case 3

Note:

The parameter $\mathbf{a} = (m + \psi_\nu^n)\mathbf{c}^{-1}$.

- The value of $\mathbf{a}$ is a polynomial with different coefficients in $\mathbb{Z}_q$, being less suspicious for the participants.

Case 3

Note:

The parameter $\mathbf{a} = (m + \psi_\nu^n)\mathbf{c}^{-1}$.

- The value of $\mathbf{a}$ is a polynomial with different coefficients in $\mathbb{Z}_q$, being less suspicious for the participants.
- We have $m = \mathbf{a}.\mathbf{c} - \psi_\nu^n$ where:
  - $\mathbf{a}$ and m are public (for m we can use brute force).
  - The value ψ_ν^n is an error (small) polynomial and the value $\mathbf{c}$ is unknown.

Case 3

Note:

The parameter $\mathbf{a} = (m + \psi_\nu^n)\mathbf{c}^{-1}$.

- The value of $\mathbf{a}$ is a polynomial with different coefficients in $\mathbb{Z}_q$, being less suspicious for the participants.
- We have $m = \mathbf{a}.\mathbf{c} - \psi_\nu^n$ where:
  - $\mathbf{a}$ and m are public (for m we can use brute force).
  - The value ψ_ν^n is an error (small) polynomial and the value $\mathbf{c}$ is unknown.

It looks like the Search Ring-LWE problem where $\mathbf{c}$ is the secret. Therefore Alice and Bob do not have knowledge about $\mathbf{a}$ and its possibility of leaking information.

Experiments

Experiments

- It was executed on a processor Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz with 3 Mb of cache and 8 GB of DDR4 Memory
- The code is available online at `https://github.com/reynaldocv/sbseg2022`.
- The experiments were executed using the value $m = 722$.

Experiments

- It was executed on a processor Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz with 3 Mb of cache and 8 GB of DDR4 Memory
- The code is available online at `https://github.com/reynaldocv/sbseg2022`.
- The experiments were executed using the value $m = 722$.

	Case 2: $\mathbf{a} = m(\psi_\mu^n)^{-1}$				
Value of μ	1	2	4	8	16
Recovered complete keys	10000	10000	9284	144	0
Avg. recovered coefficients (%)	100.0	100.0	99.9	99.5	95.5
Max. # of wrong coefficients	0	0	2	19	90

	Case 3: $\mathbf{a} = (m + \psi_\nu^n)\mathbf{c}^{-1}$				
Value of $\mu = \nu$	1	2	4	8	16
Recovered complete keys	10000	9372	9	0	0
Avg. recovered coefficients (%)	100.0	99.9	99.5	95.4	84.2
Max. # of wrong coefficients	0	3	17	88	218

Table: Results of experiments

Concluding remarks

Concluding remarks

- **"How to know when the value of the public global parameter a may or may not leak information about the secret?"**

Concluding remarks

- **“How to know when the value of the public global parameter a may or may not leak information about the secret?”**

In our experiments for cases 2 and 3, the generated values **a** that leak information always have at least 10 repeated coefficients.

References

Alkim, E., Avanzi, R. M., Bos, J. W., Ducas, L., de la Piedra, A., Pöppelmann, T., and Schwabe, P. (2017).
Newhope algorithm specifications and supporting documentation.

Barreto, P. S., Longa, P., Naehrig, M., Ricardini, J. E., and Zanon, G. (2016).
Sharper ring-lwe signatures.
Cryptology ePrint Archive.

Bos, J. W., Costello, C., Naehrig, M., and Stebila, D. (2015).
Post-quantum key exchange for the tls protocol from the ring learning with errors problem.
In 2015 IEEE Symposium on Security and Privacy, pages 553–570. IEEE.

de Clercq, R., Roy, S. S., Vercauteren, F., and Verbauwhede, I. (2015).
Efficient software implementation of ring-lwe encryption.
In Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, DATE '15, page 339?344, San Jose, CA, USA. EDA Consortium.

Fan, J. and Vercauteren, F. (2012).
Somewhat practical fully homomorphic encryption.
Cryptology ePrint Archive.

References

Lindner, R. and Peikert, C. (2011).
Better key sizes (and attacks) for lwe-based encryption.
In Kiayias, A., editor, Topics in Cryptology – CT-RSA 2011, pages 319–339, Berlin, Heidelberg. Springer Berlin Heidelberg.

Lyubashevsky, V., Peikert, C., and Regev, O. (2013).
On ideal lattices and learning with errors over rings.
J. ACM, 60(6).

Peikert, C. (2009).
Public-key cryptosystems from the worst-case shortest vector problem: Extended abstract.
In Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, STOC '09, page 333?342, New York, NY, USA. Association for Computing Machinery.

Regev, O. (2009).
On lattices, learning with errors, random linear codes, and cryptography.
volume 56, New York, NY, USA. Association for Computing Machinery.

Roy, S. S., Karmakar, A., and Verbauwhede, I. (2016).
Ring-lwe: applications to cryptography and their efficient realization.
In International conference on security, privacy, and applied cryptography engineering, pages 323–331. Springer.

References

Wu, Y., Huang, Z., Zhang, J., and Wen, Q. (2012).
A lattice-based digital signature from the ring-lwe.
In 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content, pages 646–651.