TTM4135 Worksheet 9: Post-Quantum Cryptography

Tjerand Silde and Emil August Hovd Olaisen
{tjerand.silde, emil.august.olaisen}@ntnu.no

Spring 2026

This exercise is an extension of Worksheet 7, with additional tasks about post-quantum cryptography. The tasks dive deeper into the mathematics of lattice-based cryptography. In particular, we explore two schemes: a public-key encryption scheme closely resembling ML-KEM / CRYSTALS-Kyber, and a digital signature algorithm closely resembling ML-DSA / CRYSTALS-Dilithium.

The tasks are somewhat more demanding. If you struggle, feel free to skip certain parts. The questions are designed so that you can still use results from previous tasks you have not completed. This worksheet is intended to help you better understand lattice-based cryptography, but it will not be directly relevant for the exam.

Outline:

- Task 1: properties of the polynomial ring $R_{q,d}$,
- Task 2: a module-based public-key encryption scheme,
- Task 3: communication costs,
- Task 4: computational efficiency,
- Task 5: motivation for using modules,
- Task 6: a module-based digital signature scheme.

The ring $R_{q,d}$. An element $a \in R_{q,d}$ is a vector

$$a = (a_0, a_1, \ldots, a_{d-1}), \quad a_i \in \mathbb{Z}_q.$$

Addition is component-wise. Multiplication is defined using matrix multiplication:

$$a \cdot b = \begin{bmatrix} a_0 & -a_{d-1} & \ldots & -a_1 \\ a_1 & a_0 & \ldots & -a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{d-1} & a_{d-2} & \ldots & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{d-1} \end{bmatrix}.$$

Equivalently,

$$(a \cdot b)_i = \sum_{j=0}^{i} a_{i-j} b_j - \sum_{j=i+1}^{d-1} a_{d+i-j} b_j.$$

You may assume the following properties without proof:

$$a \cdot b = b \cdot a, \tag{1}$$
$$a \cdot (b \cdot c) = (a \cdot b) \cdot c, \tag{2}$$
$$a \cdot (b + c) = a \cdot b + a \cdot c. \tag{3}$$

Thus $(R_{q,d}, +, \cdot)$ is a commutative ring. Vectors and matrices over $R_{q,d}$ behave exactly as over the integers. We write $\mathbf{s} \in [\beta]^n$ to mean that each coefficient of every entry $s_i \in R_{q,d}$ lies in $\{-\beta, \ldots, \beta\}$.

1. (a) Show that computing $a \cdot b$ for $a, b \in R_{q,d}$ requires d^2 multiplications and $d(d-1)$ additions in $\mathbb{Z}_q$.
   (b) Let $s, e \in [\beta]$. Show that $se \in [d\beta^2]$.
   (c) Let $\mathbf{s}, \mathbf{e} \in [\beta]^n$. Show that $\mathbf{se} \in [dn\beta^2]$.

Solution:

(a) From the definition of multiplication in $R_{q,d}$, each output coefficient $(a \cdot b)_i$ is computed as a sum of d products $a_j b_k$ in $\mathbb{Z}_q$. Since there are d output coefficients, this results in d^2 multiplications in $\mathbb{Z}_q$.

Each coefficient sum contains $d-1$ additions, giving a total of $d(d-1)$ additions in $\mathbb{Z}_q$.

(b) Let $s, e \in [\beta]$. Each coefficient of se is a sum of at most d products of coefficients bounded by β in absolute value. Each product is bounded by β^2, hence $|(se)_i| \le d\beta^2$, so $se \in [d\beta^2]$.

(c) The inner product $\mathbf{se} = \sum_{i=1}^{n} s_i e_i$ consists of n products in $R_{q,d}$. Each product lies in $[d\beta^2]$ by Part (b), so summing n such elements yields

$$\mathbf{se} \in [dn\beta^2].$$

Public-key encryption over modules. We use $R_{q,d}$ to encrypt messages $M \in \{0,1\}^d$ by interpreting M as a polynomial with binary coefficients.

KGen: sample $\mathbf{s}, \mathbf{e}_1 \leftarrow\$ [\beta]^n$ and output

$$\mathsf{sk} = \mathbf{s}, \qquad \mathsf{pk} = (\mathbf{A} \leftarrow\$ R_{q,d}^{n \times n},\ \mathbf{t} = \mathbf{As} + \mathbf{e}_1).$$

$\mathsf{Enc}(\mathsf{pk}, M)$: sample $\mathbf{r}, \mathbf{e}_2 \leftarrow\$ [\beta]^n$, $e_3 \leftarrow\$ [\beta]$, and output

$$\mathbf{u}^\intercal = \mathbf{r}^\intercal \mathbf{A} + \mathbf{e}_2^\intercal, \quad v = \mathbf{r}^\intercal \mathbf{t} + e_3 + M\lfloor q/2 \rfloor.$$

$\mathsf{Dec}(\mathsf{sk}, c)$: output $M' = v - \mathbf{u}^\intercal \mathbf{s}$.

2. (a) Show that
$$M' = \mathbf{r}^\intercal \mathbf{e}_1 - \mathbf{e}_2^\intercal \mathbf{s} + e_3 + M\lfloor q/2 \rfloor.$$
   (b) Show that $M' = e + M\lfloor q/2 \rfloor$ with
$$e \in [2dn\beta^2 + \beta].$$
   (c) Explain why each coefficient of M can be recovered correctly if
$$2dn\beta^2 + \beta < \frac{q}{4}.$$
   (d) Define an explicit decoding function
$$\mathsf{Decode} : \mathbb{Z}_q \to \{0,1\},$$
   and explain how it is applied coefficient-wise to recover M from M'.

Solution:

(a) By definition,

$$M' = v - \mathbf{u}^\intercal \mathbf{s}.$$

Substituting the encryption equations gives

$$\begin{aligned} M' &= \mathbf{r}^\intercal(\mathbf{As} + \mathbf{e}_1) + e_3 + M\lfloor q/2 \rfloor - (\mathbf{r}^\intercal \mathbf{A} + \mathbf{e}_2^\intercal)\mathbf{s} \\ &= \mathbf{r}^\intercal \mathbf{e}_1 - \mathbf{e}_2^\intercal \mathbf{s} + e_3 + M\lfloor q/2 \rfloor. \end{aligned}$$

(b) By Task 1, $\mathbf{r}^\intercal \mathbf{e}_1$ and $\mathbf{e}_2^\intercal \mathbf{s}$ lie in $[dn\beta^2]$. Together with $e_3 \in [\beta]$, we obtain

$$e = \mathbf{r}^\intercal \mathbf{e}_1 - \mathbf{e}_2^\intercal \mathbf{s} + e_3 \in [2dn\beta^2 + \beta].$$

(c) Each coefficient of M' has the form

$$M'_i = e_i + M_i \lfloor q/2 \rfloor.$$

If $|e_i| < q/4$, then values near 0 decode to $M_i = 0$ and values near $\lfloor q/2 \rfloor$ decode to $M_i = 1$. Thus correct decryption is guaranteed if

$$2dn\beta^2 + \beta < q/4.$$

(d) A standard decoding function is defined as

$$\mathsf{Decode}(x) = \begin{cases} 0 & \text{if } x \in \left(-\frac{q}{4}, \frac{q}{4}\right) \bmod q, \\ 1 & \text{if } x \in \left(\frac{q}{4}, \frac{3q}{4}\right) \bmod q. \end{cases}$$

The decrypted polynomial $M' \in R_{q,d}$ has coefficients

$$M'_i = e_i + M_i \lfloor q/2 \rfloor.$$

Applying Decode independently to each coefficient recovers M_i, provided $|e_i| < q/4$.

3. (a) How many bits are required to store elements of $R_{q,d}$, $R_{q,d}^n$, and $R_{q,d}^{n \times n}$? Use this to compute the sizes of the public key, secret key, and ciphertext.

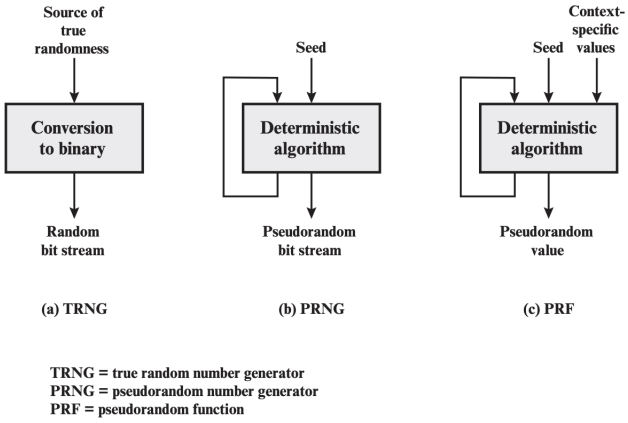

Figure 8.1 Random and Pseudorandom Number Generators

Figure 1: Generating random values from a seed using a TRNG, PRNG, and PRF.

(b) Recall Figure 1. Explain how the public key can be compressed by transmitting a seed instead of $\mathbf{A}$. Why can this not be done for $\mathbf{t}$? What about the secret key and ciphertext?

(c) Assume $\mathsf{LWE}_{n_1,q,\beta}$ and $\mathsf{MLWE}_{n_2,d,q,\beta}$ are equally hard when $n_1 = n_2 d$. Explain how increasing d and decreasing n affects the transmission size and message length.

Solution:

(a) An element of $R_{q,d}$ consists of d coefficients in $\mathbb{Z}_q$, requiring $d \log_2 q$ bits.

Thus:

$$|R^n_{q,d}| = nd \log_2 q, \qquad |R^{n\times n}_{q,d}| = n^2 d \log_2 q.$$

Hence:

- Public key: $\mathbf{A}$ and $\mathbf{t}$ are of $(n^2+n)d \log_2 q$ bits (since it is uniform)
- Secret key: $\mathbf{s}$ is of $nd \log_2(2\beta+1)$ bits (since it is bounded)
- Ciphertext: $(\mathbf{u}, v)$ is of $(n+1)d \log_2 q$ bits (since it is uniform)

(b) Matrix $\mathbf{A}$ can be generated deterministically from a uniformly random seed using a PRNG, so the public key needs to transmit only the seed. The matrix can then be reconstructed by the receiver.

The vector $\mathbf{t}$ cannot be compressed this way, since it depends on the secret vector $\mathbf{s}$ and noise $\mathbf{e}_1$.

The secret key $\mathbf{s}$ can be generated from a seed and then recomputed every time it is needed for decryption.

The ciphertext $(\mathbf{u}, v)$ contains fresh encryption randomness and message-dependent structure. While encryption internally uses a seed to derive randomness, this seed must be freshly and uniformly random for each encryption. Transmitting such a seed would make ciphertexts deterministic or linkable and would therefore break semantic (IND-CPA) security.

(c) If $n_1 = n_2 d$ yields equal hardness, then increasing d and decreasing n reduces key and ciphertext sizes while increasing the plaintext length from 1 bit to d bits. Thus transmission cost per plaintext bit decreases.

4. (a) Count the number of multiplications in $\mathbb{Z}_q$ required to compute:
   - $\mathbf{ab}$ for $\mathbf{a}, \mathbf{b} \in R^n_{q,d}$,
   - $\mathbf{Ab}$ for $\mathbf{A} \in R^{n\times n}_{q,d}$.

   (b) Count the number of vector and matrix-vector multiplications in KGen, Enc, and Dec.

   (c) What is the total cost (in $\mathbb{Z}_q$ multiplications) per bit of M?

Solution:

(a) Each multiplication in $R_{q,d}$ costs d^2 multiplications in $\mathbb{Z}_q$. Thus:

- $\mathbf{ab}$ requires nd^2 multiplications in $\mathbb{Z}_q$.
- $\mathbf{Ab}$ requires $n^2 d^2$ multiplications in $\mathbb{Z}_q$.

(b)
- KGen: one matrix–vector multiplication
- Enc: one matrix–vector multiplication and one inner product
- Dec: one inner product

(c) Total complexity is dominated by $O(n^2 d^2)$ operations. Since each ciphertext conveys d bits, the cost per bit is $O(n^2 d)$.

5. Let $\beta = 1$ and $q = 3329$. Assume $\mathsf{LWE}_{602,q,\beta}$ is secure and we wish to exchange a 256-bit key.

   (a) Explain why choosing $n = 3$ and $d = 256$ is reasonable.

   (b) Show that correct decryption is not unconditionally guaranteed.

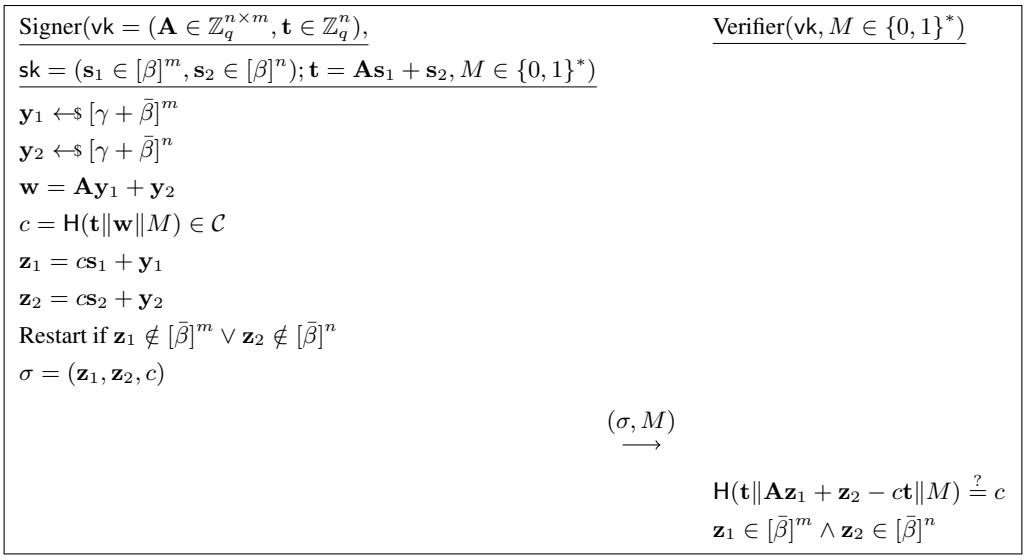

Figure 2: A lattice-based Schnorr-type signature.

(c) Estimate the probability that decryption fails.

Solution:

(a) Choosing $n = 3$ and $d = 256$ preserves the effective dimension $n \cdot d = 768$, exceeding the assumed secure LWE dimension of 602, while allowing fast NTT-based arithmetic and compact ciphertexts.

(b) From Task 2, correctness requires

$$2dn\beta^2 + \beta < q/4.$$

With $d = 256$, $n = 3$, $\beta = 1$, the left-hand side equals 1537, while $q/4 \approx 832$, violating the condition. Hence decryption may fail with (potentially) non-negligible probability.

(c) Let e_i denote a coefficient of the effective noise

$$e = \mathbf{r}^\intercal \mathbf{e}_1 - \mathbf{e}_2^\intercal \mathbf{s} + e_3.$$

Decryption fails for coefficient i if $|e_i| \geq q/4$. While an exact probability cannot be computed without specifying the noise distribution, we can reason heuristically.

Each e_i is a sum of approximately $2dn + 1$ independent bounded random variables with small support. By the Central Limit Theorem, the distribution of e_i is close to a discrete Gaussian with variance proportional to dn.

Since $2dn\beta^2 + \beta > q/4$, the correctness bound is violated, and there is a (potentially) non-negligible but small probability that $|e_i| \geq q/4$. However, because $q = 3329$ is relatively large and $\beta = 1$, this probability is still small in practice.

6. Consider the lattice-based Schnorr-type signature scheme in Figure 2.

(a) Show that the signature scheme is correct when it does not reject the signature.

(b) What is the size of the challenge space $\mathcal{C}$, defined as the subset of polynomials in $[1]^d$ in which η elements are non-zero, and how does this impact security?

(c) Explain why $\mathbf{z}_1$ and $\mathbf{z}_2$ are bounded before rejection and by what values.

Solution:

(a) Using $\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$, we compute:

$$\mathbf{A}\mathbf{z}_1 + \mathbf{z}_2 - c\mathbf{t} = \mathbf{A}\mathbf{y}_1 + \mathbf{y}_2 = \mathbf{w}.$$

Thus the verifier recomputes the same hash challenge c, so verification accepts.

(b) The challenge space $\mathcal{C}$ consists of vectors with exactly η non-zero ± 1 entries, giving

$$|\mathcal{C}| = \binom{d}{\eta} 2^{\eta}.$$

A larger challenge space increases security by lowering the probability that an adversary can guess a valid challenge in advance. This directly impacts the soundness of the signature scheme and its resistance to forgery, in the same way as the challenge size in the classical Schnorr signature.

(c) Before rejection, each coefficient $|z_i| \leq \eta\beta + \bar{\beta} + \gamma$, since $\mathbf{z}_i = c\mathbf{s}_i + \mathbf{y}_i$ and c has η non-zero entries.