Lattice-based Post-Quantum Cryptography
Number Theoretic Transform

Bhumika Mittal
Department of Computer Science
Ashoka University

Abstract: The Number Theoretic Transform (NTT) is a crucial function in many post-quantum cryptographic schemes based on lattices like dilithium and kyber. This presentation discusses the basic theory of NTT and some of its variants.

Lattice Basics

- **Lattices**: For any $m \in \mathbb{N}$, an m-dimensional lattice L is a discrete additive subgroup of $(\mathbb{R}^m, +)$, where $\mathbb{R}^m$ is the m-dimensional Euclidean space.
- **An Alternate Definition**: Let $\mathcal{L} \subseteq \mathbb{R}^m$. Then, $\mathcal{L}$ is a lattice if and only if there exist k linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_k \in \mathcal{L}$ such that

$$\mathcal{L} = \left\{ \sum_{i=1}^{k} c_i \mathbf{b}_i \;\middle|\; c_i \in \mathbb{Z} \text{ and } \mathbf{b}_1, \ldots, \mathbf{b}_k \in \mathbb{R}^m \right\}$$

- The integer k is called the **rank** of the lattice. Clearly, $k \leq m$. The sequence of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_k$ is called a **lattice basis** and it is conveniently represented as a matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_k] \in \mathbb{R}^{m \times k}$.
- Using the matrix notation, the above can be written in a more compact form as $L = \{\mathbf{Bc} \,|\, \mathbf{c} \in \mathbb{Z}^k\}$, where $\mathbf{Bc}$ is the usual matrix-vector multiplication.
- When $m = k$, the lattice is called **full-rank**.
- An m-dimensional lattice $\mathcal{L}$ is called an **integer lattice** if $\mathcal{L} \subseteq \mathbb{Z}^m$.

Lattice Basics

- **Lattice Determinant** Let $\mathcal{L} = \mathcal{L}(B)$ be a full-rank lattice. Define $\det(\mathcal{L}) = |\det(B)|$.
- **Minimum Distance**

$$\lambda_1 = \min_{\substack{x,y \in \mathcal{L} \\ x \neq y}} \|\mathbf{x} - \mathbf{y}\|$$

$$= \min_{\substack{x \in \mathcal{L} \\ x \neq 0}} \|\mathbf{x}\|$$

- **Successive minima** $(1 \leq i \leq m)$

$$\lambda_i = \min\{r \mid \dim(\text{span}_R(\mathcal{B}(r) \cap \mathcal{L})) \geq i\}$$

- When $\mathcal{L} = \mathbb{Z}^m$, we have $\lambda_1 = \lambda_2 = \ldots = \lambda_m = 1$.
- Fact $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_m$

Learning With Errors (LWE)

- Sampling from $\mathsf{LWE}_{n,m,q,\alpha}$

$\text{Sample } \{(a_i, b_i)\}_{i=1}^{m} \xleftarrow{\mathsf{LWE}_{n,m,q,\alpha}} \mathbb{Z}_q^n \times \mathbb{Z}_q$
$s \leftarrow \mathbb{Z}_q^n$
For $i = 1$ to m
$\quad a_i \leftarrow \mathbb{Z}_q^{n \times 1}$
$\quad e_i \leftarrow \mathbb{Z}$, where $\sigma = \alpha q$
$\quad b_i = a_i^T s + e_i \pmod q$
Output $\{(a_i, b_i)\}_{i=1}^{m}$

- Another representation for $\{(a_i, b_i)\}_{i=1}^{m} \equiv (A \in \mathbb{Z}_q^{n \times m}, b \in \mathbb{Z}_q^m)$, where $A = \begin{bmatrix} a_1 & \cdots & a_m \end{bmatrix}$ and $b = A^T s + e \pmod q$ where $e = \begin{bmatrix} e_1 & \cdots & e_m \end{bmatrix}^T$.

- **Decision-$\mathsf{LWE}_{n,m,q,\alpha}$**
  Given $\{(a_i, b_i)\}_{i=1}^{m}$, decide between

$$\{(a_i, b_i)\}_{i=1}^{m} \xleftarrow{\mathsf{LWE}_{n,m,q,\alpha}} \mathbb{Z}_q^n \times \mathbb{Z}_q$$

$$\{(a_i, b_i)\}_{i=1}^{m} \xleftarrow{\$} \mathbb{Z}_q^n \times \mathbb{Z}_q$$

- **Search-$\mathsf{LWE}_{n,m,q,\alpha}$**
  Given $\{(a_i, b_i)\}_{i=1}^{m} \xleftarrow{\mathsf{LWE}_{n,m,q,\alpha}} \mathbb{Z}_q^n \times \mathbb{Z}_q$, determine s.

- *Remark*: There exists various algorithms that solves Decison-LWE and the efficiency of these algorithms depends upon the range of the security parameter α

Polynomial Multiplication

- Input $\left| \begin{array}{l} f(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} \\ g(x) = b_0 + b_1 x + \cdots + b_{n-1} x^{n-1}, \ a_i, b_i \in \mathbb{R}, \text{ a ring.} \end{array} \right.$

  Output $f(x) \times g(x) = c_0 + c_1 x + \cdots + c_{2n-2} x^{2n-2}$, where $c_j = \sum_{k=0}^{j} a_k b_{j-k}$, $0 \leq j \leq 2n-1$

- Define $R_q = \frac{\mathbb{Z}_q[x]}{\langle x^n - 1 \rangle}$, q is prime.

  Input $\left| \begin{array}{l} f(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} \\ g(x) = b_0 + b_1 x + \cdots + b_{n-1} x^{n-1}, \ f, g \in R_q \end{array} \right.$

  Output $f \cdot g (\text{mod } R_q)$ where
  $f \cdot g (\text{mod } R_q) = f \times g (\text{mod } q, x^n - 1) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$

- Define $R_q = \frac{\mathbb{Z}_q[x]}{\langle x^n + 1 \rangle}$, q is prime.

  Input $\left| \begin{array}{l} f(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} \\ g(x) = b_0 + b_1 x + \cdots + b_{n-1} x^{n-1}, \ f, g \in R_q \end{array} \right.$

  Output $f \cdot g (\text{mod } R_q)$ where
  $f \cdot g (\text{mod } R_q) = f \times g (\text{mod } q, x^n + 1) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$

Scheme	Ring	q	n
Kyber	$\frac{\mathbb{Z}_q[x]}{\langle x^n+1\rangle}$	3329	256
Dilithium	$\frac{\mathbb{Z}_q[x]}{\langle x^n+1\rangle}$	$2^{23} - 2^{13} + 1 = 8380417$	256

Table: Polynomial Multiplications in NIST PQC Finalists

Number Theoretic Transform

- **n-th Primitive Root Modulo q**: An n-th primitive root modulo q is a $\omega \in \mathbb{Z}_q^*$ such that the order of ω (with respect to the group $\mathbb{Z}_q^*$) is n, i.e., $\omega^n \equiv 1 \pmod{q}$ and for any $m < n$, $\omega^m \not\equiv 1 \pmod{q}$.
- Fact A prime q admits n-th primitive root of unity $\Leftrightarrow q \equiv 1 \pmod{n}$.
  Notation: ω_n - for nth primitive root.
- **$\mathrm{NTT}_{\omega_n}(f)$** Let $f(x) \in \mathbb{Z}_q[x]$. Define

$$\mathrm{NTT}_{\omega_n}(f) = (f(1), f(\omega_n), \ldots, f(\omega_n^{n-1}))) \in \mathbb{Z}_q^n$$

  where $f(\omega_i^n)$s are evaluated modulo q.
- **Lemma** Let $f, g \in \frac{\mathbb{Z}_q[x]}{\langle x^n - 1 \rangle}$ and ω_n be an nth primitive root modulo q. Then,

$$f \cdot g \,(\mathrm{mod}\, q, x^n - 1) = (n^{-1}) \cdot \mathrm{NTT}_{\omega_n^{-1}}(\mathrm{NTT}_{\omega_n}(f) \cdot \mathrm{NTT}_{\omega_n}(g))$$

  where $\mathrm{NTT}_{\omega_n}(f) \cdot \mathrm{NTT}_{\omega_n}(g) = (f(1) \cdot g(1), f(\omega_n) \cdot g(\omega_n), \ldots, f(\omega_n^{n-1})) \cdot g(\omega_n^{n-1}))$ and (n^{-1}) is computed modulo q.

$$\mathrm{NTT}_w(f) = \begin{bmatrix} f(1) \\ f(\omega) \\ \vdots \\ f(\omega^{n-1}) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 + \cdots + a_{n-1} \\ \vdots \\ a_0 + a_1\omega^{n-1} + \cdots + a_{n-1}(\omega^{n-1})^{n-1} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \omega^{n-1} & (\omega^{n-1})^2 & \cdots & (\omega^{n-1})^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

$$= V(\omega) \cdot c(f)$$

Fact Since $V_{\omega_n}^{-1}$ exists in $\mathbb{Z}_q$, and $V_{\omega_n}^{-1} = n^{-1} \cdot V_{\omega_n^{-1}}$, therefore:

$$\begin{aligned} \mathrm{cof}(f) &= V_{\omega_n}^{-1} \cdot \mathrm{NTT}_{\omega_n}(f) \\ &= n^{-1} \cdot V_{\omega_n^{-1}} \cdot \mathrm{NTT}_{\omega_n}(f) \\ &= V_{\omega_n^{-1}} \cdot n^{-1} \cdot \mathrm{NTT}_{\omega_n}(f) \\ &= \mathrm{NTT}_{\omega_n^{-1}}(n^{-1} \cdot \mathrm{NTT}_{\omega_n}(f)) \end{aligned}$$

Example

For the following parameters:

$$q = 7$$
$$n = 3, n^{-1} = 5$$
$$\omega = 2, \omega^{-1} = 4$$
$$f(x) = 1 + 2x + x^2$$

We get the following conversion using the direct method

$$c(f) = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \xrightarrow{\text{NTT}(f)} \begin{bmatrix} f(1) \\ f(2) \\ f(4) \end{bmatrix} = \begin{bmatrix} 1+2+1 = 4 \mod 7 \\ 1+4+4 = 9 \mod 7 \\ 1+8+16 = 25 \mod 7 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 4 \end{bmatrix} \xrightarrow{\text{NTT}} \begin{bmatrix} 5.4 = 20 \implies 6 \\ 5.2 = 10 \implies 3 \\ 5.4 = 20 \implies 6 \end{bmatrix}$$

This means, we have $g(x) = 6 + 3x + 6x^2$

$$\begin{aligned} \text{NTT}_{\omega^{-1}}(g) &= (g(1), g(\omega), g(\omega^2)) \\ &= (g(1), g(4), g(2)) \\ &= (1, 2, 1) \end{aligned}$$

Computing $\mathrm{NTT}_{\omega_n}(f)$: Algorithm 1

Assume $n = 2^k$

- Let $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$.
- Write $f(x) = f_e(x^2) + x \cdot f_o(x^2)$, where
  $f_e(x^2) = a_0 + a_2 x^2 + a_4 x^4 + \cdots + a_{n-2} x^{n-2} = a_0 + a_2 x^2 + a_4 (x^2)^2 + \cdots + a_{n-1} (x^2)^{\frac{n}{2}-1}$
  $f_o(x^2) = a_1 + a_3 x^2 + a_5 x^4 + \cdots + a_{n-1} x^{n-2} = a_1 + a_3 x^2 + a_5 (x^2)^2 + \cdots + a_{n-1} (x^2)^{\frac{n}{2}-1}$
- $NTT_{\omega_n}(f) = (f(\omega_n^i))_{i=0}^{n-1}$. Therefore, for $1 \leq i \leq n$,

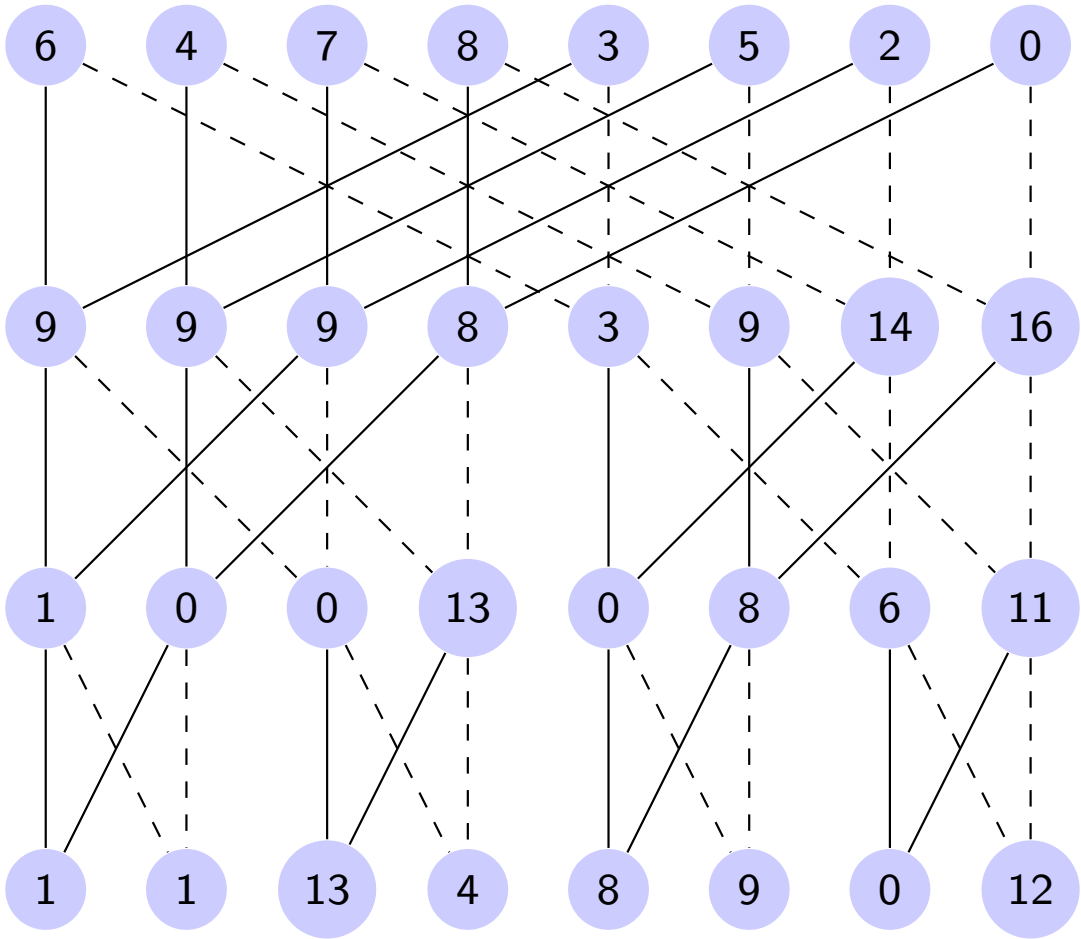
$$\begin{aligned} f(\omega_n^i) &= f_e((\omega_n^i)^2) + \omega_n^i \cdot f_o((\omega_n^i)^2) \\ &= f_e((\omega_{2n})^i) + \omega_n^i \cdot f_o((\omega_{2n})^i) \\ &= f_e((\omega_{n/2})^i) + \omega_n^i \cdot f_o((\omega_{n/2})^i) \end{aligned}$$

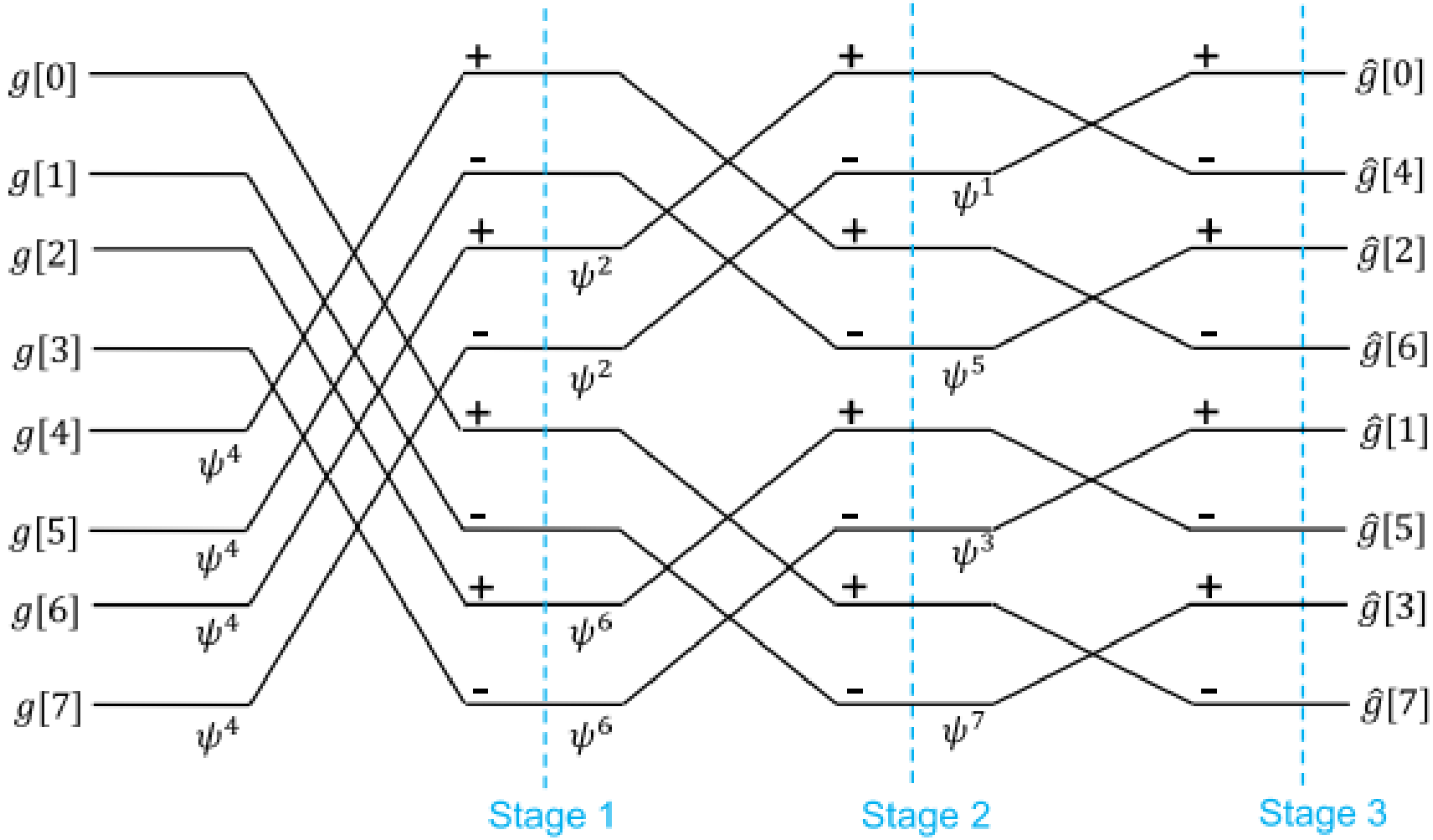
  where $\omega_{n/2} = \omega_n^2$ is $n/2$-th primitive root modulo q.
- Therefore, for $0 \leq i \leq \frac{n}{2} - 1$, $f(\omega_n^i) = f_e((\omega_{n/2})^i) + \omega_n^i \cdot f_o((\omega_{n/2})^i)$, and for $\frac{n}{2} + 1 \leq i \leq n - 1$, write $i = \frac{n}{2} + j$, where $0 \leq j \leq \frac{n}{2} - 1$, and therefore
  $f(\omega_n^i) = f(\omega_n^{\frac{n}{2}+j}) = f_e((\omega_{n/2})^{\frac{n}{2}+j}) + \omega_n^{\frac{n}{2}+j} \cdot f_o((\omega_{n/2})^{\frac{n}{2}+j}) = f_e((\omega_{n/2})^j) - \omega_n^j \cdot f_o((\omega_{n/2})^j)$
- Running Time $T(n) = 2T(n/2) + \Theta(n) = \Theta(n \log n)$

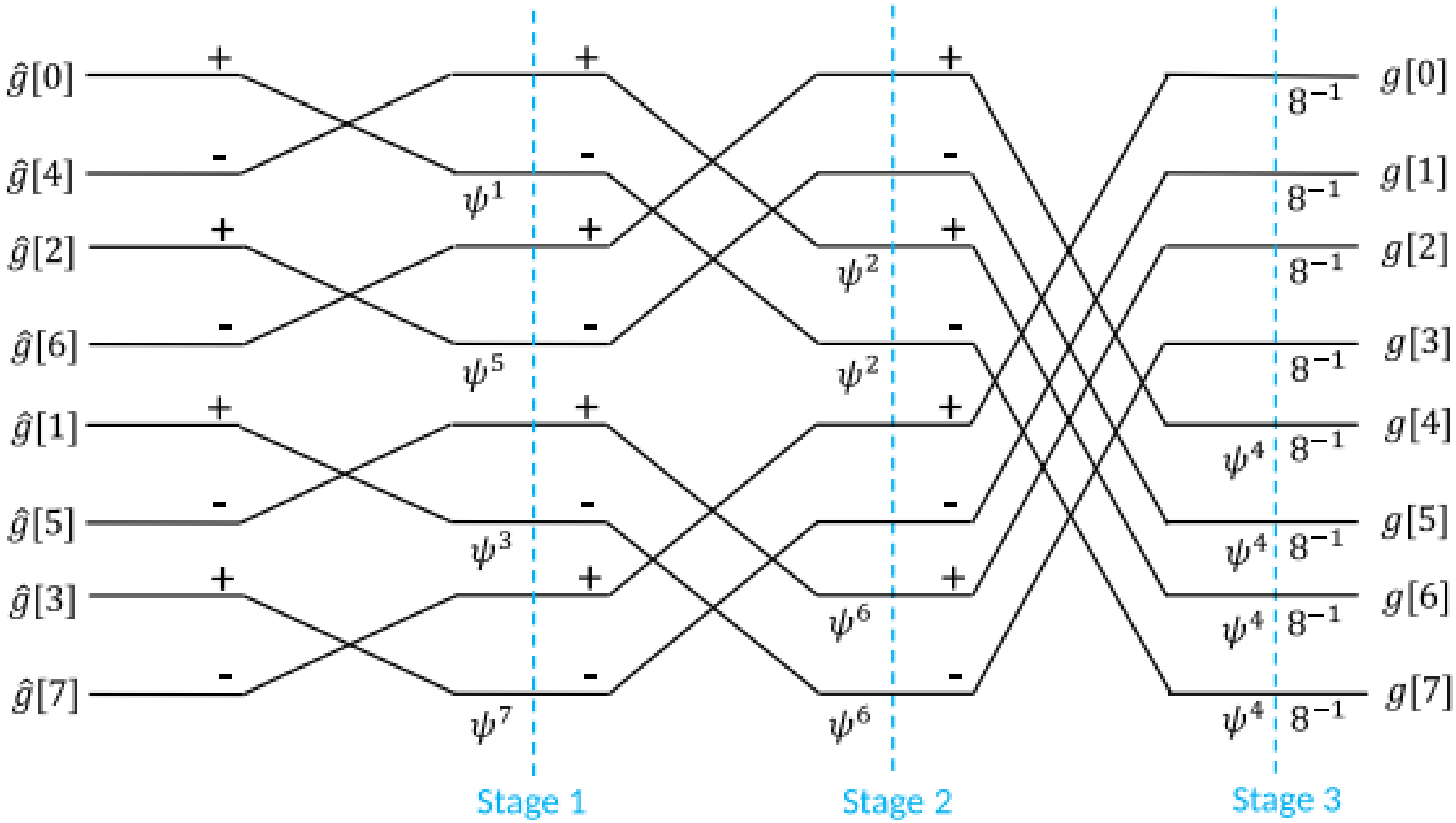
Butterfly diagram

Example

Let $n = 8$, $q = 17$




$g[0]$
$g[1]$
$g[2]$
$g[3]$
$g[4]$
$g[5]$
$g[6]$
$g[7]$
ψ^4
ψ^4
ψ^4
ψ^4
ψ^2
ψ^2
ψ^6
ψ^6
ψ^1
ψ^5
ψ^3
ψ^7
$\hat{g}[0]$
$\hat{g}[4]$
$\hat{g}[2]$
$\hat{g}[6]$
$\hat{g}[1]$
$\hat{g}[5]$
$\hat{g}[3]$
$\hat{g}[7]$
Stage 1
Stage 2
Stage 3


$\hat{g}[0]$
$\hat{g}[4]$
$\hat{g}[2]$
$\hat{g}[6]$
$\hat{g}[1]$
$\hat{g}[5]$
$\hat{g}[3]$
$\hat{g}[7]$
+
-
ψ^1
ψ^5
ψ^3
ψ^7
ψ^2
ψ^6
ψ^4
8^{-1}
$g[0]$
$g[1]$
$g[2]$
$g[3]$
$g[4]$
$g[5]$
$g[6]$
$g[7]$
Stage 1
Stage 2
Stage 3

Multiplication in $\frac{\mathbb{Z}_q[x]}{\langle x^n+1\rangle}$

- Consider $R_q = \frac{\mathbb{Z}_q[x]}{\langle x^n+1\rangle}$ where $n = 2^k$, $k \in \mathbb{N}$, and q is a prime such that $q \equiv 1 \pmod{2n}$. As q is prime, $\mathbb{Z}_q$ is a field, and since $q \equiv 1 \pmod{2n}$, $\mathbb{Z}_q^*$ contains an element ω with $o(\omega) = 2n$ (the $2n$th primitive root of unity).
- Let $H = \langle \omega \rangle \subseteq \mathbb{Z}_q^*$ be the subgroup generated by ω.
- The total number of generators of $H = \varphi(2n) = \varphi(2^{k+1}) = 2^{k+1} - 2^k = 2^k = n$. These generators (primitive roots) are clearly the odd powers of ω, i.e., they are $\omega, \omega^3, \omega^5, \ldots, \omega^{2n-1}$.
- Also, all elements of H are roots of the polynomial $x^{2n} - 1$ over $\mathbb{Z}_q$. As $x^{2n} - 1 = (x^n - 1) \times (x^n + 1)$, for any generator ω_0 of H, we have $((\omega_0)^n - 1) \times ((\omega_0)^n + 1) = (\omega_0)^{2n} - 1 = 0$. This means, $(\omega_0)^{n+1} = 0$ ($(\omega_0)^n - 1 = 0$ contradicts the primitiveness). Therefore, all generators of H are roots of the polynomial $x^n + 1$. Therefore, we have

$$x^n + 1 = \prod_{i=1,3,\ldots,2n-1} (x - \omega^i)$$

- This means (using the Chinese Remainder Theorem), $\mathbb{Z}_q[x]/\langle x^n+1\rangle$ is isomorphic to $\frac{\mathbb{Z}_q[x]}{\langle x-\omega\rangle} \times \frac{\mathbb{Z}_q[x]}{\langle x-\omega^3\rangle} \times \ldots \times \frac{\mathbb{Z}_q[x]}{\langle x-\omega^{2n-1}\rangle}$.
- The isomorphism is given by:

$$f(x) \in \mathbb{Z}_q[x]/\langle x^{n+1}\rangle \mapsto (f(\omega), f(\omega^3), \ldots, f(\omega^{2n-1}))$$

- Therefore, for any two $f, g \in R_q$ we have

$$\begin{aligned} f \cdot g \pmod{R_q} &= I^{-1}(I(f) \cdot I(g)) \\ &= I^{-1}(\mathrm{DFT}_{\omega_2}(f_0) \cdot \mathrm{DFT}_{\omega_2}(g_0)) \\ &= I^{-1}(\mathrm{DFT}_{\omega_2}(f_0 \cdot g_0)) \\ &= I^{-1}(I(f \cdot g)) \\ &= \boxed{(1, \omega^{-1}, \omega^{-2}, \ldots, \omega^{-255}) \cdot \left(\frac{1}{n} \cdot \mathrm{DFT}(\omega^2)^{-1}(I(f \cdot g))\right)} \end{aligned}$$

NTT in Dilithium

- Ring: $R = \frac{\mathbb{Z}_q[x]}{<x^n+1>}$
- q is chosen such that there exists $2n$-th root of unity (mod q)
- $q = 2^{23} - 2^{13} + 1 = 8380417$
- $n = 256$
- $\psi = 1753$

The Zetas

There are some pre-computed values in the dilithium code, namely, zetas. For $\psi = 1753$ and $q = 8380417$, we have the following:

$$\begin{aligned}\mathbb{Z}_q &= \{0, 1, \cdots, 8380416\} \\ &= \{-\frac{8380416}{2}, \cdots, 0, \cdots, \frac{8380416}{2}\}\end{aligned}$$

We will define,

$$zeta[i] = \psi^{br(i)} \times 2^{32} \mod q$$

If

$$zeta[i] > \frac{q-1}{2}, zeta[i] = zeta[i] - q$$

Remark: We are multiplying by 2^{32} due to montgomery reduction which is used to carry out the multiplication efficiently and prevent overflowing.

Why is NTT nice?

- Both NTT and INTT are linear transformations, based on which it can save INTTs in Dilithium, Kyber, and other lattice-based schemes.

$$\sum_{i=0}^{n} a_i b_i = \sum_{i=0}^{n} \mathrm{INTT}(\mathrm{NTT}(a_i) \circ \mathrm{NTT}(b_i))$$
$$= \mathrm{INTT}\left(\sum_{i=0}^{n} \mathrm{NTT}(a_i) \circ \mathrm{NTT}(b_i)\right)$$

- Consider $c = \mathrm{INTT}(\mathrm{NTT}(a) \circ \mathrm{NTT}(b))$, where a is random. Since the NTT transforms keep the randomness of a random polynomial, i.e., $\hat{a} = \mathrm{NTT}(a)$ is also random, one can directly generate a random polynomial, and view it as random $\hat{a}$ already in the NTT domain, and compute $c = \mathrm{INTT}(\hat{a} \circ \mathrm{NTT}(b))$, thus eliminating the need for the forward transform.

Remark: This is done in Dilithium.

Thank you!

Acknowledgement: Inspired by the discussions with Professor Mahavir Jhawar on this topic