The NTT and residues of a polynomial modulo factors of $X^{2^d} + 1$

Sahil Sharma

Signify Research, Eindhoven, The Netherlands
sahil.sharma@signify.com

Abstract. The Number Theoretic Transform (NTT) plays a central role in efficient implementations of cryptographic primitives selected for Post Quantum Cryptography. Although it certainly exists, academic papers that cite the NTT omit the connection between the NTT and residues of a polynomial modulo factors of $X^{2^d}+1$ and mention only the final expressions of what the NTT computes. This short paper establishes that connection and, in doing so, elucidates key aspects of computing the NTT. Based on this, the specific instantiations of the NTT function used in CRYSTALS-Kyber and CRYSTALS-Dilithium are derived.

Keywords: NTT, Kyber, Dilithium, Post Quantum Cryptography (PQC), Efficient implementations of PQC

1 Introduction

The NIST [NIS] selected Post Quantum Cryptography algorithms Dilithium [DKL+18] and Kyber [BDK+18] perform polynomial multiplication in $\mathbb{Z}_q[X]/(X^{2^d}+1)$, where prime q is such that a primitive $2^{(n+1)}$-th root of unity exists. Both use the NTT [Ber01] to perform this efficiently and the Kyber specification [ABD+21](Page 5) formulates the expressions that the NTT computes. However, there is no clear explanation on how those expressions are connected with residues of a polynomial modulo factors of $X^{2^d}+1$. In the first two sections (2 and 3), the link between the NTT and residues of a polynomial modulo factors of $X^{2^d}+1$ is established. Based on this, the specific NTT functions used in Kyber and Dilithium are formulated and the paper concludes by describing why and how the NTT is used in the efficient implementations of Kyber and Dilithium.

2 Decomposition of $X^{2^d} + 1$ into factors

Let ζ be a primitive $2^{(n+1)}$-th root of unity $\mod$ prime q. Thus, $\zeta^{2^{(n+1)}} \equiv 1 \pmod q$ and $\zeta^{2^n} \equiv -1 \pmod q$. Hence, $X^{2^d}+1 \equiv X^{2^d} - \zeta^{2^n} \pmod q$. This provides a way of factoring $X^{2^d}+1$ as shown in figure 1 (shown for Kyber, for which $d=8$). In general,

$$\begin{aligned} X^{2^{d-l}} - \zeta^{s_l} &= \left(X^{2^{d-(l+1)}} - \zeta^{\frac{s_l}{2}}\right)\left(X^{2^{d-(l+1)}} + \zeta^{\frac{s_l}{2}}\right) \\ &\equiv \left(X^{2^{d-(l+1)}} - \zeta^{2^n \cdot \mathbf{0} + \frac{s_l}{2}}\right)\left(X^{2^{d-(l+1)}} - \zeta^{2^n \cdot \mathbf{1} + \frac{s_l}{2}}\right) \pmod q \left[\text{since } \zeta^{2^n} \equiv -1 \pmod q\right] \end{aligned}$$

Let $i = i_1 i_2 \ldots i_l$ be the bit sequence representing the path, starting from the top, taken to a particular s_l (as part of $X^{2^{d-l}} - \zeta^{s_l}$, refer figure 1). For example, 0100 is the sequence for $s_l = 40$ at $l = 4$. Then, in the equation above, the first factor is the immediate left

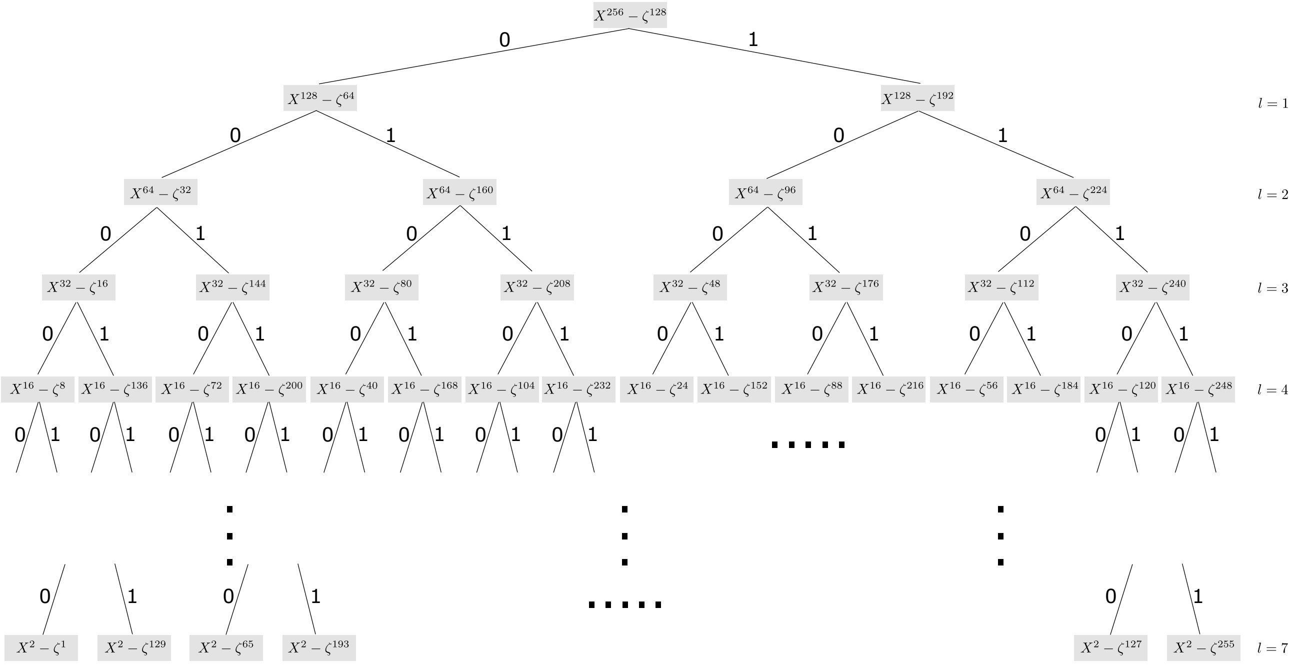


Figure 1: Decomposition of $X^{2^8}+1$ into factors (for CRYSTALS-Kyber)

lower node (edge labelled bit **0**) and the second factor is the immediate right lower node (edge labelled bit **1**). Thus, s_l's in the terms $X^{2^{d-l}}-\zeta^{s_l}$ at any layer l follow the pattern:

$$\begin{aligned} s_0 =& 2^n \\ s_{l+1} =& 2^n i_{l+1} + \frac{s_l}{2} \end{aligned} \tag{1}$$

Therefore,

$$\begin{aligned} s_1 =& 2^n i_1 + 2^{n-1} \\ s_2 =& 2^n i_2 + 2^{n-1} i_1 + 2^{n-2} \\ & ... \\ s_l =& 2^n i_l + 2^{n-1} i_{l-1} + 2^{n-2} i_{l-2} + ... 2^{n-(l-1)} i_1 + 2^{n-l} \\ =& 2^{n-(l-1)}(2^{l-1} i_l + 2^{l-2} i_{l-1} + 2^{l-3} i_{l-2} + ... i_1) + 2^{n-l} \\ =& 2^{n+1-l} BR_l(i) + 2^{n-l} \end{aligned} \tag{2}$$

where $1 \leq l \leq n$ and $BR_l(i)$ is the l-bit bit-reversal of $i = i_1 i_2 ... i_l$.

3 Reducing a polynomial modulo factors of $X^{2^d}+1$

Theorem 1. *Given a polynomial* $a \in \mathbb{Z}_q[X]/(X^{2^d}+1)$ *and a factor* $X^{2^{d-l}}-\zeta^{s_l}$ *(notations as from the previous section), the coefficient of* X^c *in the polynomial* $a \mod (X^{2^{d-l}}-\zeta^{s_l})$ *(i.e.* $\in \mathbb{Z}_q[X]/(X^{2^{d-l}}-\zeta^{s_l})$*), where* $0 \leq c \leq (2^{d-l}-1)$ *is given by*

$$\sum_{j=0}^{2^l-1} a_{2^{d-l}(j)+c} \zeta^{s_l j} \tag{3}$$

Proof. Polynomial a is of the form

$$a = a_0 + a_1X + a_2X^2 + ... + a_{2^d-1}X^{2^d-1}$$

As shown before, $X^{2^d}+1 \equiv X^{2^d}-\zeta^{2^n} \equiv \left(X^{2^{d-1}} - \zeta^{2^{(n-1)}}\right)\left(X^{2^{d-1}} - \zeta^{2^n+2^{(n-1)}}\right) \pmod{q}$. Start by splitting the polynomial into two halves, a_L, i.e. $a \mod (X^{2^{d-1}} - \zeta^{2^{(n-1)}})$, and a_R, i.e. $a \mod (X^{2^{d-1}} - \zeta^{(2^n+2^{(n-1)})})$.

$$\begin{aligned} s_1 \text{ for } a_L &= 2^{(n-1)} \text{ (from equation (2), setting } l=1 \text{ and } i=0) \\ s_1 \text{ for } a_R &= 2^n + 2^{(n-1)} \text{ (from equation (2), setting } l=1 \text{ and } i=1) \end{aligned}$$

Since $X^{2^{d-1}} \equiv \zeta^{s_1} \mod (X^{2^{d-1}} - \zeta^{s_1})$, a_L and a_R (each with its respective s_1) are

$$\begin{aligned} a_{u\in\{L,R\}} =&(a_0 + \zeta^{s_1}a_{2^{d-1}}) + (a_1 + \zeta^{s_1}a_{2^{d-1}+1})X + ... \\ &+(a_{2^{d-1}-2} + \zeta^{s_1}a_{2^d-2})X^{2^{d-1}-2} + (a_{2^{d-1}-1} + \zeta^{s_1}a_{2^d-1})X^{2^{d-1}-1} \end{aligned}$$

[Spr20] illustrates the first level of splitting a polynomial. The theorem statement is true for $l = 1$, as can be verified in the expressions for a_L, a_R. Let this statement be true for $l \geq 1$. Observe that, due to the structure of the factors in figure 1, step l produces $a \mod (X^{2^{d-l}} - \zeta^{s_l})$ (refer Appendix A.3) and there are 2^l s_l's, each resulting in a residue. Splitting each of these residue polynomials can be continued until step $l = n$ (refer figure 1). When splitting for step $l+1$, since $X^{2^{d-(l+1)}} \equiv \zeta^{s_{l+1}} \mod (X^{2^{d-(l+1)}} - \zeta^{s_{l+1}})$, coefficient of $X^{c'}$, where $0 \leq c' \leq (2^{d-(l+1)} - 1)$, in the resulting reduced polynomial is

$$\text{Coefficient of } X^{c'} \text{ from step } l + (\text{Coefficient of } X^{c'+2^{d-(l+1)}} \text{ from step } l)\zeta^{s_{l+1}}$$

From equation (3), which is true for l by the induction hypothesis, this becomes

$$\begin{aligned} &\sum_{j=0}^{2^l-1} a_{2^{d-l}(j)+c'}\zeta^{s_lj} + \left(\sum_{j=0}^{2^l-1} a_{2^{d-l}(j)+2^{d-(l+1)}+c'}\zeta^{s_lj}\right)\zeta^{s_{l+1}} \\ =&\sum_{j=0}^{2^l-1} a_{2^{d-l}(j)+c'}\zeta^{s_lj} + \left(\sum_{j=0}^{2^l-1} a_{2^{d-(l+1)}(2j+1)+c'}\zeta^{s_lj}\right)\zeta^{s_{l+1}} \\ =&\sum_{j=0}^{2^l-1} a_{2^{d-(l+1)}(2j)+c'}\zeta^{\frac{s_l}{2}2j} + \left(\sum_{j=0}^{2^l-1} a_{2^{d-(l+1)}(2j+1)+c'}\zeta^{\frac{s_l}{2}2j}\right)\zeta^{s_{l+1}} \end{aligned} \tag{4}$$

Note that s_l is even for $1 \leq l \leq (n-1)$ (i.e. until the last step in this mathematical induction) and, thus, $\frac{s_l}{2}$ is an integer. Using the fact that ζ is a $2^{(n+1)}$-th primitive root of unity mod q in equation (1),

$$\zeta^{s_{l+1}2j} = \zeta^{(2^ni_{l+1}+\frac{s_l}{2})2j} = \zeta^{2^{(n+1)}i_{l+1}j}\zeta^{\frac{s_l}{2}2j} \equiv \zeta^{\frac{s_l}{2}2j} \pmod{q} \tag{5}$$

Using equation (5) in equation (4),

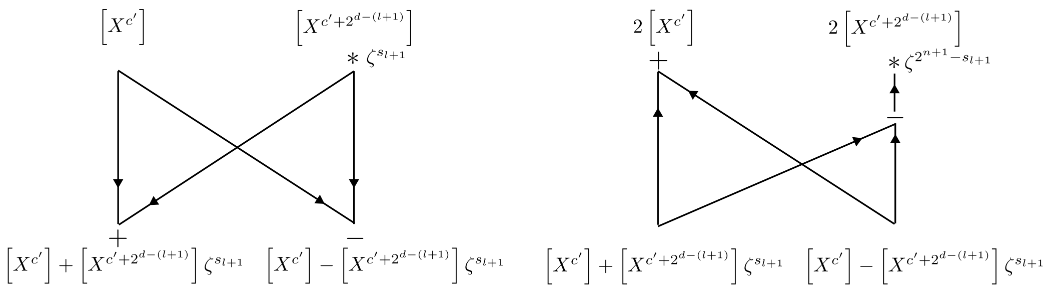
$$\begin{aligned} &\sum_{j=0}^{2^l-1} a_{2^{d-(l+1)}(2j)+c'}\zeta^{s_{l+1}2j} + \sum_{j=0}^{2^l-1} a_{2^{d-(l+1)}(2j+1)+c'}\zeta^{s_{l+1}(2j+1)} \\ =&\sum_{j=0}^{2^l-1}\left(a_{2^{d-(l+1)}(2j)+c'}\zeta^{s_{l+1}2j} + a_{2^{d-(l+1)}(2j+1)+c'}\zeta^{s_{l+1}(2j+1)}\right) \\ =&\sum_{j'=0}^{2^{l+1}-1} a_{2^{d-(l+1)}(j')+c'}\zeta^{s_{l+1}j'} \end{aligned}$$

Thus, equation (3) is also valid for $l+1$. □

Corollary 1. *It should be noted that the theorem is equally valid for factors of* $X^{2^d}-1$*.* $X^{2^d}-\zeta^0 \equiv \left(X^{2^{d-1}}-\zeta^0\right)\left(X^{2^{d-1}}-\zeta^{2^n}\right) \pmod q$*. Thus,* $s_0=0$ *in equation (1), which gives* $s_l = 2^{n+1-l}BR_l(i)$ *and the steps proceed exactly as proven above. This formulation ((3) with* s_l*'s shown here) is referred to as the cyclic NTT. An elegant relationship between these, resulting in what is known as twisting, is explained in [Ber07].*

Corollary 2. *It can be observed, both from figure 1 as well as the steps taken in the theorem, that at step* $l+1$*, the number of polynomials mod* $\left(X^{2^{d-(l+1)}}-\zeta^{s_{l+1}}\right)$*, each of which is computed with a particular* s_{l+1} *(see equation (4)), is twice that of the number in step* l*. However, each of these polynomials has half the number of terms as compared to the polynomial it was split from in step* l*. Therefore, the total number of multiplications to compute polynomials at any step remains constant (*2^d*). Since there are a total of* n *steps and* $n \leq d$*, the total number of multiplications is at most* $d2^d$*. Furthermore, from equation (4), it can be seen that, when splitting for step* $l+1$*, after multiplication with a particular* s_{l+1}*, an addition is performed, implying that the total number of additions required is the same as the total number of multiplications, i.e. at most* $d2^d$*. Finally, note that most papers use the notation* X^n+1*, instead of* $X^{2^d}+1$ *used here, and the number of multiplications/additions, expressed with this notation, is* $n\log_2 n$*.*

Corollary 3. *For the same reason, the total number (number of polynomials* $*$ *terms in each polynomial) of terms remains constant at each step* l*. Implementations use this property (after reduction modulo* q*) to perform the above computation in-place.*


$\left[X^{c'}\right]$
$\left[X^{c'+2^{d-(l+1)}}\right]$
$*\ \zeta^{s_{l+1}}$
$+$
$-$
$\left[X^{c'}\right]+\left[X^{c'+2^{d-(l+1)}}\right]\zeta^{s_{l+1}}$
$\left[X^{c'}\right]-\left[X^{c'+2^{d-(l+1)}}\right]\zeta^{s_{l+1}}$
$2\left[X^{c'}\right]$
$2\left[X^{c'+2^{d-(l+1)}}\right]$
$+$
$*\ \zeta^{2^{n+1}-s_{l+1}}$
$-$
$\left[X^{c'}\right]+\left[X^{c'+2^{d-(l+1)}}\right]\zeta^{s_{l+1}}$
$\left[X^{c'}\right]-\left[X^{c'+2^{d-(l+1)}}\right]\zeta^{s_{l+1}}$

Forward (Cooley-Tukey) butterfly   Inverse (Gentleman-Sande) butterfly

Figure 2: The Cooley-Tukey [CT65] and Gentleman-Sande [GS66] butterflies

Figure 2 (where $[X^t]$ denotes the coefficient of X^t) illustrates the structure (known as a butterfly) that results from the operation performed in equation (4). Implementations use this structure to compute residues of a polynomial modulo degree 1/degree 2 factors of $X^{2^d}+1$. The decomposition of a polynomial into a set of such residues is what the NTT (strictly speaking, equation (3) with s_l's as defined in equation (2) is known as the Negacyclic NTT) of a polynomial computes. Note that, in practice, when splitting a polynomial at step $l+1$, a single multiplication is enough to compute both resulting polynomials (i.e. the total multiplications mentioned in corollary 2 is halved to $\frac{d}{2}2^d$). This is because s_{l+1}'s for the two polynomials differs by 2^n (refer equation (1)). Thus, $\zeta^{s_{l+1}}$ for $a'_R = (\zeta^{s_{l+1}}$ for $a'_L)\,\zeta^{2^n} \equiv -\zeta^{s_{l+1}}$ for $a'_L \pmod q$ (as shown in the forward butterfly in figure 2).
The inverse butterfly (Inverse NTT or NTT^{-1}) reverses the operation of step $l+1$ to produce coefficients of the polynomial (i.e. the one at step l) that was split in two. Notice that each inversion step produces $2\left[X^{c'}\right]$ and $2\left[X^{c'+2^{d-(l+1)}}\right]$. This is corrected by multiplying all the coefficients by 2^{-n} at the end of the inverse computation.

4 NIST selected Post Quantum Cryptographic algorithms

NIST [NIS] has selected Kyber for Public-key Encryption and Key-establishment and Dilithium for Digital Signatures as part of its effort towards Post-Quantum Cryptography. The NTT plays a central role in the efficient implementations of both Kyber and Dilithium. The following subsections illustrate the application of equation (3) to these.

4.1 The NTT function for Kyber

For Kyber [ABD+21] $d = 8$, the prime $q = 3329$. Since $q - 1 = 2^8 \cdot 13$, it has 256-th primitive roots of unity and, thus, $n = 7$. Setting $n = 7$ in equation (2), the s_l 's at $l = 7$ are

$$2BR_7(i) + 1 \text{ , where } 0 \leq i \leq 127$$

From equation (3), the residues, $a \mod (X^2 - \zeta^{(2BR_7(i)+1)})$ are

$$\left(\sum_{j=0}^{127} a_{2j}\zeta^{(2BR_7(i)+1)j}\right) + \left(\sum_{j=0}^{127} a_{2j+1}\zeta^{(2BR_7(i)+1)j}\right) X \tag{6}$$

The term on the left (coefficient of X^0) is denoted as $\hat{a}_{2i}$ and the coefficient of X is denoted as $\hat{a}_{2i+1}$. Expression (6) is what is mentioned in [ABD+21](Page 5). Thus, the residues are

$$(\hat{a}_0 + \hat{a}_1 X, \hat{a}_2 + \hat{a}_3 X, ...\hat{a}_{254} + \hat{a}_{255} X)$$

The function, which takes input coefficients $(a_0, a_1, ..., a_{255})$ of a polynomial a and produces $(\hat{a}_0, \hat{a}_1, ..., \hat{a}_{2i}, \hat{a}_{2i+1}, ..., \hat{a}_{255})$, as expressed in (6), is the NTT function for Kyber, denoted as $\mathbf{NTT}(a)$. This can also be written out in matrix form as

$$\text{Let } \mathbf{N} = \begin{pmatrix} 1 & \zeta^{1\cdot 1} & \zeta^{1\cdot 2} & \cdots & \zeta^{1\cdot 127} \\ 1 & \zeta^{3\cdot 1} & \zeta^{3\cdot 2} & \cdots & \zeta^{3\cdot 127} \\ 1 & \zeta^{5\cdot 1} & \zeta^{5\cdot 2} & \cdots & \zeta^{5\cdot 127} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^{255\cdot 1} & \zeta^{255\cdot 2} & \cdots & \zeta^{255\cdot 127} \end{pmatrix}$$

Then,

$$\begin{pmatrix} \hat{a}_0 \\ \hat{a}_{2BR_7(1)} \\ \hat{a}_{2BR_7(2)} \\ \vdots \\ \hat{a}_{2BR_7(127)} \end{pmatrix} = \mathbf{N} \begin{pmatrix} a_0 \\ a_2 \\ \vdots \\ a_{254} \end{pmatrix} \text{ and } \begin{pmatrix} \hat{a}_1 \\ \hat{a}_{2BR_7(1)+1} \\ \hat{a}_{2BR_7(2)+1} \\ \vdots \\ \hat{a}_{2BR_7(127)+1} \end{pmatrix} = \mathbf{N} \begin{pmatrix} a_1 \\ a_3 \\ \vdots \\ a_{255} \end{pmatrix}$$

Notice above that the output coefficients are in bit-reversed order, which is due to the presence of $BR_l(i)$ in the expression for s_l 's. Notice, also, that the NTT matrix, $\mathbf{N}$, is similar to a Discrete Fourier Transform (DFT) matrix (the NTT is, in fact, a variation of the DFT defined over a finite field). As would be expected, $\mathbf{N}$ has an inverse, given by

$$\mathbf{N}^{-1} = 2^{-7} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ \zeta^{-1\cdot 1} & \zeta^{-3\cdot 1} & \zeta^{-5\cdot 1} & \cdots & \zeta^{-255\cdot 1} \\ \zeta^{-1\cdot 2} & \zeta^{-3\cdot 2} & \zeta^{-5\cdot 2} & \cdots & \zeta^{-225\cdot 2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \zeta^{-1\cdot 127} & \zeta^{-3\cdot 127} & \zeta^{-5\cdot 127} & \cdots & \zeta^{-255\cdot 127} \end{pmatrix}$$

It can easily be shown that $\mathbf{N}^{-1}\mathbf{N} = \mathbf{I}$.

4.2 The NTT function for Dilithium

For Dilithium [BDK+21], $d = 8$, prime $q = 8380417$. Since $q - 1 = 2^{13} \cdot 3 \cdot 11 \cdot 31$, it has 512-th primitive roots of unity and, thus, $n = 8$. Setting $n = 8$ in equation (2), the s_l 's at $l = 8$ are

$$2BR_8(i) + 1 \text{ , where } 0 \leq i \leq 255$$

From equation (3), the residues, $a \mod (X - \zeta^{(2BR_8(i)+1)})$ are

$$\sum_{j=0}^{255} a_j \zeta^{(2BR_8(i)+1)j} \tag{7}$$

The function, which takes input coefficients $(a_0, a_1, ..., a_{255})$ of a polynomial a and produces $(\hat{a}_0, \hat{a}_1, ..., \hat{a}_i, ..., \hat{a}_{255})$, as expressed in (7), is the NTT function for Dilithium, denoted as $\mathbf{NTT}(a)$. This can also be written out in matrix form as

$$\text{Let } \mathbf{N} = \begin{pmatrix} 1 & \zeta^{1\cdot1} & \zeta^{1\cdot2} & \cdots & \zeta^{1\cdot255} \\ 1 & \zeta^{3\cdot1} & \zeta^{3\cdot2} & \cdots & \zeta^{3\cdot255} \\ 1 & \zeta^{5\cdot1} & \zeta^{5\cdot2} & \cdots & \zeta^{5\cdot255} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^{511\cdot1} & \zeta^{511\cdot2} & \cdots & \zeta^{511\cdot255} \end{pmatrix} \text{ Then, } \begin{pmatrix} \hat{a}_0 \\ \hat{a}_{BR_8(1)} \\ \hat{a}_{BR_8(2)} \\ \vdots \\ \hat{a}_{BR_8(255)} \end{pmatrix} = \mathbf{N} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{255} \end{pmatrix}$$

The inverse of $\mathbf{N}$ is

$$\mathbf{N}^{-1} = 2^{-8} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ \zeta^{-1\cdot1} & \zeta^{-3\cdot1} & \zeta^{-5\cdot1} & \cdots & \zeta^{-511\cdot1} \\ \zeta^{-1\cdot2} & \zeta^{-3\cdot2} & \zeta^{-5\cdot2} & \cdots & \zeta^{-511\cdot2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \zeta^{-1\cdot255} & \zeta^{-3\cdot255} & \zeta^{-5\cdot255} & \cdots & \zeta^{-511\cdot255} \end{pmatrix}$$

5 Properties of the NTT

This section lists important properties of the NTT.

5.1 NTT of the addition of two polynomials

Given two polynomials a and b,

$$\mathbf{NTT}\,(a \pm b) = \mathbf{NTT}(a) \pm \mathbf{NTT}(b)$$

From equation (3), the coefficient of X^c in the polynomial $a \pm b \mod (X^{2^{d-l}} - \zeta^{s_l})$ (i.e. $\in \mathbb{Z}_q[X]/(X^{2^{d-l}} - \zeta^{s_l})$), where $0 \leq c \leq (2^{d-l} - 1)$ is given by

$$\sum_{j=0}^{2^l-1} \left(a_{2^{d-l}(j)+c} \pm b_{2^{d-l}(j)+c}\right) \zeta^{s_l j}$$

The result trivially follows from associative and distributive properties:

$$= \sum_{j=0}^{2^l-1} a_{2^{d-l}(j)+c} \zeta^{s_l j} \pm \sum_{j=0}^{2^l-1} b_{2^{d-l}(j)+c} \zeta^{s_l j}$$

5.2 NTT of the product of two polynomials

Given two polynomials a and b,

$$\mathbf{NTT}\,(ab) = \mathbf{NTT}(a) \circ \mathbf{NTT}(b)$$

where $\circ$ denotes basecase multiplication (refer to Appendix A). This can be proven using the Chinese Remainder Theorem [BDK+21] [Sei18], which establishes an isomorphism from the ring $\mathbb{Z}_q[X]/(X^{2^d}+1)$ to (the product of rings) $\prod_{i=0}^{127} \mathbb{Z}_q[X]/(X^2 - \zeta^{(2BR_7(i)+1)})$ for Kyber and to (the product of rings) $\prod_{i=0}^{255} \mathbb{Z}_q[X]/(X - \zeta^{(2BR_8(i)+1)})$ for Dilithium. For the sake of completeness, however, direct proofs are provided for the NTT functions used in Kyber (A.1) and Dilithium (A.2).

6 NTT and efficient implementations

The previous sections showed the matrix forms of the NTT and its inverse and drew out their similarity to the DFT. In implementations, however, they are computed using efficient algorithms. The forward computation proceeds similar to the steps shown in theorem 1 and the computational structure that arises as a result is referred to as the Cooley-Tukey (CT) [CT65] butterfly. Similarly, the inverse operation proceeds by reversing the CT butterfly and the computational structure that arises as a result is referred to as the Gentleman-Sande [GS66] butterfly. Table 1 shows a comparison of the operational counts between regular (schoolbook) and NTT based multiplication of two polynomials.

Table 1: Comparison between regular and NTT based multiplication

Operation	regular multiplication	NTT multiplication
NTT	-	$\frac{d}{2}2^d$ Mul, $d2^d$ Add (per polynomial)
Multiplication	2^{2d} Mul, $2^d\left(2^d-1\right)$ Add	Kyber: $2.5 \cdot 2^d$ Mul, 2^d Add (A.1) Dilithium: 2^d Mul (A.2)
Inverse NTT	-	$\frac{d}{2}2^d$ Mul, $d2^d$ Add
Total	2^{2d} Mul, $2^d\left(2^d-1\right)$ Add	Kyber: $\left(\frac{3}{2}d+2.5\right)2^d$ Mul, $(3d+1)\,2^d$ Add Dilithium: $\left(\frac{3}{2}d+1\right)2^d$ Mul, $3d \cdot 2^d$ Add

Overall, it is evident that, while regular polynomial multiplication has a complexity of $\mathcal{O}(2^{2d})$, NTT based multiplication has a complexity of $\mathcal{O}(d2^d)$. Both Kyber and Dilithium involve numerous polynomial multiplications and additions. In Kyber [ABD+21], for instance, during key generation, the following computation is performed

$$\mathbf{As} + \mathbf{e}$$

Here $\mathbf{A}$ is a matrix, each element of which is a polynomial (i.e. $\in \mathbb{Z}_q[X]/(X^{2^d}+1)$), while $\mathbf{s}$ and $\mathbf{e}$ are column vectors, each element of which is also a polynomial. Although the performance benefit of NTT based multiplication described above essentially demonstrates $\mathbf{NTT^{-1}}\,(\mathbf{NTT}\,(a) \circ \mathbf{NTT}\,(b))$, it is not necessary to perform the inverse NTT of the product. Further optimizations can be achieved by keeping results in the NTT domain and performing the inverse NTT only when required. Matrix $\mathbf{A}$ is generated in the NTT domain (denoted as $\hat{\mathbf{A}}$). Elements of column vectors, such as $\mathbf{s}$, are converted to the NTT domain (denoted as $\hat{\mathbf{s}}$), Then, the expression described above can efficiently be computed as

$$\hat{\mathbf{A}} \circ \hat{\mathbf{s}} + \hat{\mathbf{e}}$$

where $\circ$ denotes basecase multiplication between an element of $\hat{\mathbf{A}}$ and an element of $\hat{\mathbf{s}}$, as shown in 5.2, and addition of elements is also performed in the NTT domain, as shown in 5.1. The public key and the secret key are kept in the NTT domain in order that the en/decryption routines can directly use these without first having to convert into the NTT domain. Where necessary, an inverse NTT, which has a computational complexity similar to the NTT, is performed to convert a vector back into its non NTT form. Dilithium employs similar techniques for efficiently computing polynomial multiplication and these can be found in its specification [BDK+21].

Note that the multiplications and additions mentioned in table 1 are actually modular multiplications and additions performed $\mod$ prime q. Towards this end, implementations may represent polynomial coefficients in Montgomery form [Mon85]. Since the primes used by Kyber and Dilithium are relatively small, the result of the multiplication of any two polynomial coefficients fits in 32 and 64 bits respectively. This fact is exploited in [Sei18] to perform signed Montgomery modular arithmetic. Setting β (or R in some literature) associated with Montgomery arithmetic to $2^{16}/2^{32}$, the method uses operations that take the lower 16/32 bits of a multiplication, 16/32 bit right shifts, etc. which are efficient on 32/64 bit architectures. Plantard also uses this fact about the size of primes in the proposed Plantard arithmetic [Pla21]. A variation of this technique is described in [HZZ+22] and comparisons of the approach are made with Montgomery modular multiplication, Barrett reduction [Bar87], etc.

Observe (refer to equation (4)) that the 2^l residue polynomials at any step l can be computed (by multiplying coefficients of a particular polynomial with its associated s_l, followed by addition) in parallel. The computational structure of the NTT (and, similarly, the inverse NTT), therefore, is well suited to Single Instruction Multiple Data (SIMD) extensions such as AVX2 [Sei18], making implementations even faster.

To conclude, as further reading, the interested reader is encouraged to read literature that describe methods to use the Cooley–Tukey butterfly also for the inverse NTT [AHKS22] and the advantages of doing so.

References

[ABD+21] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation(version 3.02). Technical report, pq-crystals, 2021. URL: https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf.

[AHKS22] Amin Abdulrahman, Vincent Hwang, Matthias J. Kannwischer, and Amber Sprenkels. Faster Kyber and Dilithium on the Cortex-M4. In Giuseppe Ateniese and Daniele Venturi, editors, *Applied Cryptography and Network Security*, pages 853–871, Cham, 2022. Springer International Publishing.

[Bar87] Paul Barrett. Implementing the Rivest Shamir and Adleman Public Key Encryption Algorithm on a Standard Digital Signal Processor. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 311–323, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

[BDK+18] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367, 2018. doi:10.1109/EuroSP.2018.00032.

[BDK+21] Shi Bai, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium Algorithm Specifications and Supporting Documentation (Version 3.1). 2021. URL: https://www.pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf.

[Ber01] Daniel J. Bernstein. Multidigit multiplication for mathematicians. 09 2001.

[Ber07] Daniel J. Bernstein. The tangent fft. In Serdar Boztaş and Hsiao-Feng (Francis) Lu, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pages 291–300, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[CT65] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. URL: http://www.jstor.org/stable/2003354.

[DKL+18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(1):238?268, Feb. 2018. URL: https://tches.iacr.org/index.php/TCHES/article/view/839, doi:10.13154/tches.v2018.i1.238-268.

[GS66] W. M. Gentleman and G. Sande. Fast fourier transforms: For fun and profit. In *Proceedings of the November 7-10, 1966, Fall Joint Computer Conference*, AFIPS 6́6 (Fall), pages 563–578, New York, NY, USA, 1966. Association for Computing Machinery. doi:10.1145/1464291.1464352.

[HZZ+22] Junhao Huang, Jipeng Zhang, Haosong Zhao, Zhe Liu, Ray C. C. Cheung, Çetin Kaya Koç, and Donglong Chen. Improved Plantard Arithmetic for Lattice-based Cryptography. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, Aug. 2022. URL: https://tches.iacr.org/index.php/TCHES/article/view/9833, doi:10.46586/tches.v2022.i4.614-636.

[Mon85] Peter L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44:519–521, 1985. URL: https://api.semanticscholar.org/CorpusID:119574413.

[NIS] NIST. National Institute of Standards and Technology Post-Quantum Cryptography. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

[Pla21] Thomas Plantard. Efficient word size modular arithmetic. *IEEE Transactions on Emerging Topics in Computing*, 9(3):1506–1518, 2021. doi:10.1109/TETC.2021.3073475.

[Sei18] Gregor Seiler. Faster AVX2 optimized NTT multiplication for Ring-LWE lattice cryptography. *IACR Cryptology ePrint Archive*, 2018:39, 2018. URL: https://api.semanticscholar.org/CorpusID:4037516.

[Spr20] Amber Sprenkels. The Number Theoretic Transform in Kyber and Dilithium. https://electricdusk.com/ntt.html, September 2020.

A Appendix

The appendix contains proofs of the multiplicative property of the NTT (for both Kyber and Dilithium) and of successive splitting in 1 resulting in residues $a \mod (X^{2^{d-l}} - \zeta^{s_l})$.

A.1 NTT of ab for Kyber

Using $X^{256} \equiv -1 \mod (X^{256}+1)$, the coefficients of the odd and even powers of X in the product ab ($\in \mathbb{Z}_q[X]/(X^{256}+1)$) can be written separately. Coefficient of $X^{2j}, 0 \leq j \leq 127$ is given by

$$p_{2j} \equiv \sum_{m=0}^{127} \left(a_{2m}b^*_{2j-2m} + a_{2m+1}b^*_{2j-2m-1}\right) \mod (X^{256}+1)$$

where

$$\begin{aligned} b^*_k &= b_k \text{ if } k \geq 0 \\ &= -b_{256+k} \text{ if } k < 0 \end{aligned}$$

We also derive $\mathbf{E_m}$ and $\mathbf{O_m}$, with $m \in \{0, 1, ..., 128\}$, as

$$\begin{aligned}
\mathbf{E_m} &= \sum_{j=0}^{127} b^*_{2(j-m)}\zeta^{(2BR_7(i)+1)(j-m)} \\
&= \sum_{j=m}^{127} b_{2(j-m)}\zeta^{(2BR_7(i)+1)(j-m)} + \sum_{j=0}^{m-1} -b_{2(128+j-m)}\zeta^{(2BR_7(i)+1)(j-m)} \\
&\equiv \sum_{j=m}^{127} b_{2(j-m)}\zeta^{(2BR_7(i)+1)(j-m)} + \sum_{j=0}^{m-1} b_{2(128+j-m)}\zeta^{(2BR_7(i)+1)(128+j-m)} \pmod q \\
&= \sum_{k=0}^{127-m} b_{2k}\zeta^{(2BR_7(i)+1)(k)} + \sum_{k=128-m}^{127} b_{2k}\zeta^{(2BR_7(i)+1)(k)} = \sum_{k=0}^{127} b_{2k}\zeta^{(2BR_7(i)+1)(k)} = \hat{b}_{2i}
\end{aligned}$$

$$\begin{aligned}
\mathbf{O_m} &= \sum_{j=0}^{127} b^*_{2(j-m)+1}\zeta^{(2BR_7(i)+1)(j-m)} \\
&= \sum_{j=m}^{127} b_{2(j-m)+1}\zeta^{(2BR_7(i)+1)(j-m)} + \sum_{j=0}^{m-1} -b_{2(128+j-m)+1}\zeta^{(2BR_7(i)+1)(j-m)} \\
&\equiv \sum_{j=m}^{127} b_{2(j-m)+1}\zeta^{(2BR_7(i)+1)(j-m)} + \sum_{j=0}^{m-1} b_{2(128+j-m)+1}\zeta^{(2BR_7(i)+1)(128+j-m)} \pmod q \\
&= \sum_{k=0}^{127-m} b_{2k+1}\zeta^{(2BR_7(i)+1)(k)} + \sum_{k=128-m}^{127} b_{2k+1}\zeta^{(2BR_7(i)+1)(k)} = \sum_{k=0}^{127} b_{2k+1}\zeta^{(2BR_7(i)+1)(k)} = \hat{b}_{2i+1}
\end{aligned}$$

In the third line of the derivations of $\mathbf{E_m}$ and $\mathbf{O_m}$, the fact that $\zeta^{\text{Odd multiple of 128}} \equiv -1 \pmod q$ is used. Thus, $-\zeta^{(2BR_7(i)+1)(s)} \equiv \zeta^{(2BR_7(i)+1)(s+128)} \pmod q$. From equation (6), the term $\hat{p}_{2i}$ in the NTT of ab is

$$\begin{aligned}
&\sum_{j=0}^{127} p_{2j}\zeta^{(2BR_7(i)+1)j} \\
=&\sum_{j=0}^{127}\sum_{m=0}^{127}\left(a_{2m}b^*_{2j-2m}+a_{2m+1}b^*_{2j-2m-1}\right)\zeta^{(2BR_7(i)+1)j} \\
=&\sum_{m=0}^{127}\sum_{j=0}^{127} a_{2m}b^*_{2j-2m}\zeta^{(2BR_7(i)+1)j}+\sum_{m=0}^{127}\sum_{j=0}^{127} a_{2m+1}b^*_{2j-2m-1}\zeta^{(2BR_7(i)+1)j} \\
=&\sum_{m=0}^{127} a_{2m}\zeta^{(2BR_7(i)+1)m}\sum_{j=0}^{127} b^*_{2j-2m}\zeta^{(2BR_7(i)+1)(j-m)} \\
+&\sum_{m=0}^{127} a_{2m+1}\zeta^{(2BR_7(i)+1)m}\sum_{j=0}^{127} b^*_{2(j-m-1)+1}\zeta^{(2BR_7(i)+1)(j-m-1)}\zeta^{(2BR_7(i)+1)} \\
=&\sum_{m=0}^{127} a_{2m}\zeta^{(2BR_7(i)+1)m}\mathbf{E_m}+\sum_{m=0}^{127} a_{2m+1}\zeta^{(2BR_7(i)+1)m}\mathbf{O_{m+1}}\zeta^{(2BR_7(i)+1)} \\
=&\hat{a}_{2i}\hat{b}_{2i}+\hat{a}_{2i+1}\hat{b}_{2i+1}\zeta^{(2BR_7(i)+1)}
\end{aligned}$$

Similarly, the coefficient of $X^{2j+1}, 0 \le j \le 127$ in the product ab may be written as

$$p_{2j+1} \equiv \sum_{m=0}^{127}\left(a_{2m}b^*_{2j-2m+1}+a_{2m+1}b^*_{2j-2m}\right) \mod (X^{256}+1)$$

where b^*_k is as defined before. From equation (6), the term $\hat{p}_{2i+1}$ in the NTT of ab is

$$\begin{aligned}
&\sum_{j=0}^{127} p_{2j+1}\zeta^{(2BR_7(i)+1)j} \\
=&\sum_{j=0}^{127}\sum_{m=0}^{127}\left(a_{2m}b^*_{2j-2m+1}+a_{2m+1}b^*_{2j-2m}\right)\zeta^{(2BR_7(i)+1)j} \\
=&\sum_{m=0}^{127}\sum_{j=0}^{127} a_{2m}b^*_{2j-2m+1}\zeta^{(2BR_7(i)+1)j}+\sum_{m=0}^{127}\sum_{j=0}^{127} a_{2m+1}b^*_{2j-2m}\zeta^{(2BR_7(i)+1)j} \\
=&\sum_{m=0}^{127} a_{2m}\zeta^{(2BR_7(i)+1)m}\sum_{j=0}^{127} b^*_{2j-2m+1}\zeta^{(2BR_7(i)+1)(j-m)} \\
+&\sum_{m=0}^{127} a_{2m+1}\zeta^{(2BR_7(i)+1)m}\sum_{j=0}^{127} b^*_{2j-2m}\zeta^{(2BR_7(i)+1)(j-m)} \\
=&\sum_{m=0}^{127} a_{2m}\zeta^{(2BR_7(i)+1)m}\mathbf{O_m}+\sum_{m=0}^{127} a_{2m+1}\zeta^{(2BR_7(i)+1)m}\mathbf{E_m} \\
=&\hat{a}_{2i}\hat{b}_{2i+1}+\hat{a}_{2i+1}\hat{b}_{2i}
\end{aligned}$$

Thus, we have

$$\begin{aligned}
\hat{p}_{2i}+\hat{p}_{2i+1}X &= \left(\hat{a}_{2i}\hat{b}_{2i}+\hat{a}_{2i+1}\hat{b}_{2i+1}\zeta^{(2BR_7(i)+1)}\right)+\left(\hat{a}_{2i}\hat{b}_{2i+1}+\hat{a}_{2i+1}\hat{b}_{2i}\right)X \\
&\equiv (\hat{a}_{2i}+\hat{a}_{2i+1}X)\left(\hat{b}_{2i}+\hat{b}_{2i+1}X\right) \mod (X^2-\zeta^{(2BR_7(i)+1)})
\end{aligned}$$

Thus, each term $\hat{p}_{2i} + \hat{p}_{2i+1}X$ of the NTT of ab can be obtained by a multiplication of the corresponding terms of the NTT of a and NTT of b. This is the basecase multiplication in the NTT of Kyber and, from the expression above, it can be observed to take $\frac{5}{2}2^d$ multiplications and 2^d additions (where $d = 8$).

A.2 NTT of ab for Dilithium

Using $X^{256} \equiv -1 \mod (X^{256}+1)$, the coefficient of $X^j, 0 \le j \le 255$, in the product ab ($\in \mathbb{Z}_q[X]/(X^{256}+1)$) can be written as

$$p_j \equiv \sum_{m=0}^{255} \left(a_m b^*_{j-m}\right) \mod (X^{256}+1)$$

where

$$\begin{aligned} b^*_k &= b_k \text{ if } k \ge 0 \\ &= -b_{256+k} \text{ if } k < 0 \end{aligned}$$

From equation (7), the term $\hat{p}_i$ in the NTT of ab is

$$\begin{aligned}
&\sum_{j=0}^{255} p_j \zeta^{(2BR_8(i)+1)j} \\
=&\sum_{j=0}^{255}\sum_{m=0}^{255} \left(a_m b^*_{j-m}\right) \zeta^{(2BR_8(i)+1)j} \\
=&\sum_{m=0}^{255} a_m \zeta^{(2BR_8(i)+1)m} \sum_{j=0}^{255} b^*_{j-m} \zeta^{(2BR_8(i)+1)(j-m)} \\
=&\sum_{m=0}^{255} a_m \zeta^{(2BR_8(i)+1)m} \left(\sum_{j=m}^{255} b_{j-m} \zeta^{(2BR_8(i)+1)(j-m)} + \sum_{j=0}^{m-1} -b_{256+j-m} \zeta^{(2BR_8(i)+1)(j-m)} \right) \\
\equiv&\sum_{m=0}^{255} a_m \zeta^{(2BR_8(i)+1)m} \left(\sum_{j=m}^{255} b_{j-m} \zeta^{(2BR_8(i)+1)(j-m)} + \sum_{j=0}^{m-1} b_{256+j-m} \zeta^{(2BR_8(i)+1)(256+j-m)} \right) \quad \pmod q \\
=&\sum_{m=0}^{255} a_m \zeta^{(2BR_8(i)+1)m} \left(\sum_{k=0}^{255-m} b_k \zeta^{(2BR_8(i)+1)(k)} + \sum_{k=256-m}^{255} b_k \zeta^{(2BR_8(i)+1)(k)} \right) \\
=&\sum_{m=0}^{255} a_m \zeta^{(2BR_8(i)+1)m} \sum_{k=0}^{255} b_k \zeta^{(2BR_8(i)+1)(k)} \\
=&\hat{a}_i \hat{b}_i
\end{aligned}$$

In the derivation above, the fact that $\zeta^{\text{Odd multiple of 256}} \equiv -1 \pmod q$ is used. Thus, $-\zeta^{(2BR_8(i)+1)(s)} \equiv \zeta^{(2BR_8(i)+1)(s+256)} \pmod q$.
Thus, each term $\hat{p}_i$ of the NTT of ab can be obtained by a multiplication of the corresponding terms of the NTT of a and NTT of b. This is the basecase multiplication in the NTT of Dilithium and, from the expression above, it can be observed to take 2^d multiplications (where $d = 8$).

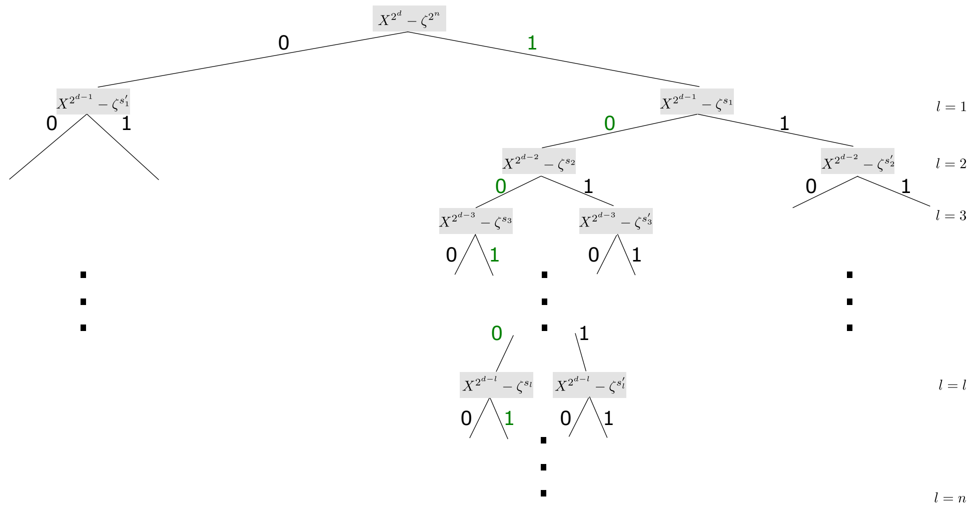

$X^{2^d} - \zeta^{2^n}$
0
1
$X^{2^{d-1}} - \zeta^{s_1'}$
0
1
$X^{2^{d-1}} - \zeta^{s_1}$
0
1
$l = 1$
$X^{2^{d-2}} - \zeta^{s_2}$
$X^{2^{d-2}} - \zeta^{s_2'}$
$l = 2$
0
1
0
1
$X^{2^{d-3}} - \zeta^{s_3}$
$X^{2^{d-3}} - \zeta^{s_3'}$
$l = 3$
0
1
0
1
0
1
$X^{2^{d-l}} - \zeta^{s_l}$
$X^{2^{d-l}} - \zeta^{s_l'}$
$l = l$
0
1
0
1
$l = n$

Figure 3: Decomposition into factors, resulting in residues $a \mod (X^{2^{d-l}} - \zeta^{s_l})$

A.3 Successive residues $a \mod (X^{2^{d-l}} - \zeta^{s_l})$

The relevant part of the structure of figure 1 for a general s_l is shown in figure 3. At $l = 1$, splitting of polynomial a produces the residue a_u, i.e. $a \mod (X^{2^{d-1}} - \zeta^{s_1})$.

$$a \equiv f_1(X)(X^{2^{d-1}} - \zeta^{s_1}) + a_u$$

At the next step, using $X^{2^{d-2}} \equiv \zeta^{s_2} \mod (X^{2^{d-2}} - \zeta^{s_2})$, a_u is reduced further as

$$\begin{aligned} a &\equiv f_1(X)(X^{2^{d-1}} - \zeta^{s_1}) + f_2(X)(X^{2^{d-2}} - \zeta^{s_2}) + a_u' \\ &\equiv f_1(X)(X^{2^{d-2}} - \zeta^{s_2'})(X^{2^{d-2}} - \zeta^{s_2}) + f_2(X)(X^{2^{d-2}} - \zeta^{s_2}) + a_u' \\ &\left[X^{2^{d-1}} - \zeta^{s_1} \equiv (X^{2^{d-2}} - \zeta^{s_2'})(X^{2^{d-2}} - \zeta^{s_2}) \pmod{q}, \text{ from decomposition into factors (figures 1, 3)}\right] \end{aligned}$$

Thus, a_u' is $a \mod (X^{2^{d-2}} - \zeta^{s_2})$. Similarly, at $l = 3$, a_u' is reduced further as

$$\begin{aligned} a \equiv &f_1(X)(X^{2^{d-2}} - \zeta^{s_2'})(X^{2^{d-3}} - \zeta^{s_3'})(X^{2^{d-3}} - \zeta^{s_3}) + f_2(X)(X^{2^{d-3}} - \zeta^{s_3'})(X^{2^{d-3}} - \zeta^{s_3}) \\ &+ f_3(X)(X^{2^{d-3}} - \zeta^{s_3}) + a_u'' \left[\text{i.e. } a_u'' \text{ is } a \mod (X^{2^{d-3}} - \zeta^{s_3})\right] \end{aligned}$$

Continuing this way, at step l:

$$\begin{aligned} a \equiv &f_1(X)(X^{2^{d-2}} - \zeta^{s_2'})(X^{2^{d-3}} - \zeta^{s_3'})...(X^{2^{d-l}} - \zeta^{s_l'})(X^{2^{d-l}} - \zeta^{s_l}) \\ &+ f_2(X)(X^{2^{d-3}} - \zeta^{s_3'})(X^{2^{d-4}} - \zeta^{s_4'})...(X^{2^{d-l}} - \zeta^{s_l'})(X^{2^{d-l}} - \zeta^{s_l}) \\ &+ ... \\ &+ f_{l-1}(X)(X^{2^{d-l}} - \zeta^{s_l'})(X^{2^{d-l}} - \zeta^{s_l}) \\ &+ f_l(X)(X^{2^{d-l}} - \zeta^{s_l}) + a_u''' \left[\text{i.e. } a_u''' \text{ is } a \mod (X^{2^{d-l}} - \zeta^{s_l})\right] \end{aligned}$$

Expressions for residues $(a_u, a_u', \text{etc.})$ is what the inductive step in theorem 1 obtains.