Main:

1. 15 ML-KEM.KeyGen()
    1.1. K-PKE.KeyGen()
        1.1.1.K-PKE.Encrypt(ek_{PKE},m,r)
            1.1.1.1. SampleNTT(XOF(ρ,i, j))
            1.1.1.2. $SamplePolyCBD_{\eta_1}$ ($PRF\eta_1$ (σ,N))
                1.1.1.2.1. BytesToBits(B)
            1.1.1.3. NTT(s)
            1.1.1.4. $ByteEncode_{12}(\hat{t})$ // ρ
                1.1.1.4.1. BitsToBytes(b)

2. ML-KEM.Encaps(ek)
    2.1. K-PKE.Encrypt(ek_{PKE},m,r)
        2.1.1. $ByteDecode_d(B)$
            2.1.1.1.1. BytesToBits(B)
        2.1.2.SampleNTT(XOF(ρ,i, j))
        2.1.3.$SamplePolyCBD_{\eta_1}$ ($PRF\eta_1$ (r,N))
            2.1.3.1.1. BytesToBits(B)
        2.1.4. NTT(r)
        2.1.5.$NTT^{-1}(\hat{f})$
        2.1.6.Decompress
        2.1.7.$ByteDecode_d$
            2.1.7.1. BytesToBits(B)
            2.1.7.2. Compress

3. ML-KEM.Decaps(c,dk)
    3.1. K-PKE.Decrypt(dk_{PKE},c)
        3.1.1.Decompress
            3.1.1.1. $ByteDecode_d(B)$
                3.1.1.1.1. BytesToBits(B)
            3.1.1.1.2.
        3.1.2.$ByteDecode_d(B)$
        3.1.3.NTT(r)
        3.1.4.$NTT^{-1}(\hat{f})$
        3.1.5.$ByteDecode_d(B)$
            3.1.5.1.1. BytesToBits(B)
        3.1.5.2.
        3.1.6.Compress

    3.2. K-PKE.Encrypt(ek_{PKE},m,r)
        3.2.1. $ByteDecode_d(B)$

3.2.1.1.1. BytesToBits(B)

3.2.2. SampleNTT(XOF(ρ,i, j))

3.2.3. SamplePolyCBD$_{\eta_1}$ (PRFη$_1$ (r,N))

3.2.3.1.1. BytesToBits(B)

3.2.4. NTT(r)

3.2.5. NTT^{-1}($\hat{f}$)

3.2.6. Decompress

3.2.7. ByteDecode$_d$

3.2.7.1. BytesToBits(B)

3.2.7.2. Compress