**Tutorial 10: A small sample on NTRU**

Instruction:

Given a random private (preferable irreducible) polynomial *f*(*x*) and a random camouflage polynomial g(*x*)

1. *f*(*x*) = −*x*10 + *x*9 + *x*6 − *x*4 + *x*2 + *x* −1 = [−1,1,0,0,1,0,−1,0,1,1,−1] (mod *p*) and
2. *g*(*x*) = −*x*10 − *x*8 + *x*5 + *x*3 + *x*2 −1 = [−1,0,−1,0,0,1,0,1,1,0,−1] in little endian.
3. an inverse *fp*−1 of *f* modulo *p*
4. and an inverse *fq*−1 of *f* modulo *q* against modulo truncated polynomial N(*x*) = *xn* −1.

*fp*−1(*x*) = 2*x*9 + *x*8 + 2*x*7 + *x*5 + 2*x*4 +2*x*3 + 2*x* +1 (mod *p*) in little endian.

= [ 2, 1, 2, 0, 1, 2, 2, 0, 2, 1] mod N(*x*)

= [-1, 1,-1, 0, 1,-1,-1, 0,-1, 1] in centered lifting format

*fq*−1(*x*) = 30*x*10 + 18*x*9 + 20*x*8 + 22*x*7 + 16*x*6 +15*x*5 + 4*x*4 +16*x*3 + 6*x*2 +9*x* +5 (mod *q*)

= [30, 18, 20, 22,16,15,4,16,6,9,5] mod N(*x*)

= [-2,-14,-12,-10,16,15,4,16,6,9,5]

Given a system parameter (*n*, p, q) = (11, 3, 32).

1. Compute a public key *h* from a given private key *f* and a blinding random g.
2. Take a plintext M= 1000 + (ID mod 1000).

M = 1000 + 39 = 1039

1. Convert M into binary.

M = 100 0000 11112

1. Convert M into a polynomial in Fp.
2. Encrypt the plaintext M using the same public key h(*x*).

*e*(*x*) = r(*x*) \**h*(*x*)

= [3,14,9,5,7,8,-4,1,0,-14,8]

1. Decrypt M back into original plaintext.

*a*(*x*)= *f*(*x*)\**e*(*x*) (modulo q)

= [29,21,5,4,1,13,18,18,2,2]

c(*x*) = *fp*−1(*x*)\**b*(*x*) (modulo p)

= [1,0,0,0,0,0,0,0,1,1,1,1]