Quantum Cyber Security
Lecture 16: Post-Quantum Cryptography III

Petros Wallden

University of Edinburgh

17th March 2026

This Lecture: NTRU Public-Key Encryption

1. Ring over Finite Field: Intro with an example
2. NTRU Public-Key Encryption: The system and its security
3. NTRU an example

Notation colour code: parameters and functions: public (blue), private (red), secret but not used later (brown)

Ring Over Finite Field: An Example

Example: Ring $R = \mathbb{Z}[x]/x^{n-1}$ (explanation below)

- Polynomials, truncated at degree n, with integer coeff $p_i \in \mathbb{Z}$: $p(x) = p_0 + p_1 x + \ldots + p_{n-1} x^{n-1}$
- Coefficients could be restricted to be in $\mathbb{Z}_q$
- The "free-parameters" characterising such polynomial are in $\mathbb{Z}_q^n$ as previously in the LWE

Parameters:

- $(n-1)$ maximum degree of polynomials. Additions of **exponents** of x are performed $\bmod n$.
- q prime number. Additions of **coefficients** (p_i's) are performed $\bmod q$

Ring Over Finite Field: An Example

- **An example of operations:** Let $n = 3$; $q = 5$.

  Consider the product of $f(x) \cdot g(x)$ in $\mathbb{Z}_5[x]/x^2$ where:

  $f(x) = 1 + 3x + 2x^2$

  $g(x) = 2 + 4x + 3x^2$

  $$\begin{aligned} f(x) \cdot g(x) &= \left(1 + 3x + 2x^2\right)\left(2 + 4x + 3x^2\right) \\ &= 2 + 4x + 3x^2 + 6x + 12x^2 + 9x^3 + 4x^2 + 8x^3 + 6x^4 \end{aligned}$$

  Exponents are taken $\mathrm{mod} 3$

  $$\begin{aligned} f(x) \cdot g(x) &= 2 + 4x + 3x^2 + 6x + 12x^2 + 9x^0 + 4x^2 + 8x^0 + 6x^1 \\ &= 19 + 16x + 19x^2 \end{aligned}$$

  Coefficients are taken $\mathrm{mod} 5$

  $$f(x) \cdot g(x) = 4 + x + 4x^2$$

NTRU Cryptosystem

- First developed in 1996 by Hoffstein, Pipher and Silverman
- Name: N(th degree) T(runcated polynomial) R(ing) U(nits)
- Both Encryption and Signatures algorithms (the former here)
- Very efficient, believed to be secure against quantum attacks
- Other versions (less efficient) have less "algebraic" structure and the hardness belief is more formally established
- No attack that uses that algebraic structure has been found (so initial version is still a valid candidate)

NTRU Encryption Scheme

Parameters: $(n-1)$ max degree of polynomials, q prime number (large mod), p prime number (small mod), d coef.

Polynomials in $\mathbb{Z}[x]/x^{n-1}$, **operations** in either $\mathbb{Z}_q[x]/x^{n-1}$ or $\mathbb{Z}_p[x]/x^{n-1}$.

Conditions on Parameters: correctness holds provided: $q > (6d+1)p$

1. KeyGen:
   - Choose two random polynomials $f(x), g(x)$ with small coefficients, that are both kept secret
   - Compute the inverses f_p^{-1}, f_q^{-1} of f w.r.t. modulo p, q: $f(x) \cdot f_p^{-1}(x) = 1 \bmod p$; $f(x) \cdot f_q^{-1}(x) = 1 \bmod q$
   - Compute $h(x) = p\left(f_q^{-1}(x) \cdot g(x)\right) \pmod q$
   - **Private Key:** $f(x), f_p^{-1}(x)$
   - **Public Key:** $h(x)$

NTRU Encryption Scheme

2. Enc($h(x), \mu$):
   - Express message μ as a polynomial $\mu(x)$ with coefficients modulo p (centred around zero).
     Example: if $p = 2$ then a n-bit message is mapped to a $(n - 1)$ degree polynomial, with $0/1$ coefficients.

NTRU Encryption Scheme

2. Enc($h(x), \mu$):
   - Express message μ as a polynomial $\mu(x)$ with coefficients modulo p (centred around zero).
     Example: if $p = 2$ then a n-bit message is mapped to a $(n-1)$ degree polynomial, with $0/1$ coefficients.
   - Randomly choose another small polynomial $r(x)$

NTRU Encryption Scheme

2. Enc($h(x), \mu$):
   - Express message μ as a polynomial $\mu(x)$ with coefficients modulo p (centred around zero).
     Example: if $p = 2$ then a n-bit message is mapped to a $(n-1)$ degree polynomial, with $0/1$ coefficients.
   - Randomly choose another small polynomial $r(x)$
   - Output $e(x) := r(x) \cdot h(x) + \mu(x) \bmod q$

NTRU Encryption Scheme

2. Enc($h(x)$, μ):
   - Express message μ as a polynomial $\mu(x)$ with coefficients modulo p (centred around zero). Example: if $p = 2$ then a n-bit message is mapped to a $(n-1)$ degree polynomial, with $0/1$ coefficients.
   - Randomly choose another small polynomial $r(x)$
   - Output $e(x) := r(x) \cdot h(x) + \mu(x) \bmod q$

3. Dec($e(x)$, $(f(x), f_p^{-1}(x))$):
   - Computes $a(x) = f(x) \cdot e(x) \pmod q$

     $a(x)$ is expressed using coefficients centred around zero, i.e. $[-q/2, q/2]$ instead of $[0, q-1]$.

NTRU Encryption Scheme

2. Enc($h(x), \mu$):
   - Express message μ as a polynomial $\mu(x)$ with coefficients modulo p (centred around zero).
     Example: if $p = 2$ then a n-bit message is mapped to a $(n-1)$ degree polynomial, with $0/1$ coefficients.
   - Randomly choose another small polynomial $r(x)$
   - Output $e(x) := r(x) \cdot h(x) + \mu(x) \bmod q$

3. Dec($e(x), (f(x), f_p^{-1}(x))$):
   - Computes $a(x) = f(x) \cdot e(x) \pmod q$

     $a(x)$ is expressed using coefficients centred around zero, i.e. $[-q/2, q/2]$ instead of $[0, q-1]$.
   - Computes $b(x) = a(x) \pmod p$

NTRU Encryption Scheme

2. Enc($h(x), \mu$):
   - Express message μ as a polynomial $\mu(x)$ with coefficients modulo p (centred around zero).
     Example: if $p = 2$ then a n-bit message is mapped to a $(n-1)$ degree polynomial, with $0/1$ coefficients.
   - Randomly choose another small polynomial $r(x)$
   - Output $e(x) := r(x) \cdot h(x) + \mu(x) \bmod q$

3. Dec($e(x), (f(x), f_p^{-1}(x))$):
   - Computes $a(x) = f(x) \cdot e(x) \pmod{q}$

     $a(x)$ is expressed using coefficients centred around zero, i.e. $[-q/2, q/2]$ instead of $[0, q-1]$.
   - Computes $b(x) = a(x) \pmod{p}$
   - Recovers message $\mu'(x) = f_p^{-1}(x) b(x) \pmod{p}$

NTRU Encryption Scheme

- **Correctness:** We consider $\mathrm{Dec}(\mathrm{Enc}(h(x), \mu), (f(x), f_p^{-1}))$.

  $a(x) = f(x) \cdot e(x) \bmod q = f(x) \cdot r(x) \cdot h(x) + f(x) \cdot \mu(x) \bmod q$

  Recall $h(x) = pf_q^{-1}(x) \cdot g(x) \bmod q$ and the first term simplifies using $f(x)f_q^{-1}(x) = 1 \bmod q$:

  $a(x) = pg(x) \cdot r(x) + f(x) \cdot \mu(x) \bmod q$

  Now $b(x) = a(x) \bmod p$ and the first term cancels (since it is multiplied by p)

  $b(x) = (f(x) \cdot \mu(x) \bmod q) \bmod p$

NTRU Encryption Scheme

Provided that $a(x)$ was centred in zero, $f(x)$ has small coefficients and $\mu(x)$ has coefficients in $[0, p-1]$ we have

$$\begin{aligned} \mu'(x) &= f_p^{-1}(x)\,(f(x) \cdot \mu(x) \bmod q) \bmod p \\ &= (f_p^{-1}(x) \cdot f(x) \cdot \mu(x)) \bmod p \\ &= \mu(x) \bmod p \end{aligned}$$

where we used $f_p^{-1}(x) \cdot f(x) = 1 \bmod p$

- **Security:** It is believed (but not proven) that the security reduces to the Closest-Vector Problem that reduces to the (approximate) SVP-problem

  A variant (SS11) is proven to reduce to approximate SVP$_\beta$

  Intuitively the $h(x) \cdot r(x)$ "masks" the message and only with the secret key one can "cancel" this term.

NTRUE: Example

Parameters: $(n, p, q, d) = (7, 3, 41, 2)$

Check: $q > (6d + 1)p$ is satisfied $41 > (6 \times 2 + 1) \times 3 = 39$

1. KeyGen:
   - $f(x) = x^6 - x^4 + x^3 + x^2 - 1$; $g(x) = x^6 + x^4 - x^2 - x$
   - $f_3^{-1}(x) = x^6 + 2x^5 + x^3 + x^2 + x + 1 \pmod 3$
   - $f_{41}^{-1}(x) = 8x^6 + 26x^5 + 31x^4 + 21x^3 + 40x^2 + 2x + 37 \pmod{41}$

     Check: $f(x) \cdot f_3^{-1}(x) = 1 \bmod 3$; $f(x) \cdot f_{41}^{-1}(x) = 1 \bmod 41$
   - **Private Key:** $f(x)$; $f_3^{-1}(x)$
   - **Public Key:** $h(x) = p\left(f_q^{-1}(x) \cdot g(x)\right) \pmod q$

     $h(x) = 20x^6 + 40x^5 + 2x^4 + 38x^3 + 8x^2 + 26x + 30 \pmod{41}$

NTRUE: Example

2. Enc($h(x), \mu = 1012202$):
   - Since $p = 3$ we need the message in ternary number. Express it as polynomial with coefficients centred around zero so $0 \to -1$, $1 \to 0$, $2 \to 1$, i.e. $1012202 \to 0, -1, 0, 1, 1, -1, 1$

     Note: if p was even, coef. not exactly centred around zero.
   - $\mu(x) = 0x^6 - 1x^5 + 0x^4 + 1x^3 + 1x^2 - 1x + 1$
   - Randomly choose: $r(x) = x^6 - x^5 + x - 1$
   - Ciphertext $e(x) := r(x) \cdot h(x) + \mu(x) \bmod q$

     $e(x) = 31x^6 + 19x^5 + 4x^4 + 2x^3 + 40x^2 + 3x + 25 \, (\mathrm{mod} 41)$

NTRUE: Example

3. $\mathrm{Dec}(e(x), f(x), f_3^{-1}(x))$
   - Compute $a(x) = f(x) \cdot e(x) \pmod{q}$

     $a(x) = x^6 + 10x^5 + 33x^4 + 40x^3 + 40x^2 + x + 40 \pmod{41}$

     which written with coefficients from $[-20, 20]$ becomes:

     $a(x) = x^6 + 10x^5 - 8x^4 - x^3 - x^2 + x - 1 \pmod{41}$
   - Compute $b(x) = a(x) \pmod{p}$

     $b(x) = x^6 + x^5 - 2x^4 - x^3 - x^2 + x - 1 \pmod{3}$
   - Recovers message: $\mu(x) = f_p^{-1}(x) b(x) \pmod{p}$

     Recall $f_3^{-1}(x) = x^6 + 2x^5 + x^3 + x^2 + x + 1$

     $\mu(x) = -x^5 + x^3 + x^2 - x + 1 \ \rightarrow\ \mu = 1012202$