Introduction to NTRU Public Key Cryptosystem[†]

NTRUEncrypt

Ling Song

May 30, 2021

[†]Credit for some slides: Hosein Hadipour

Outline

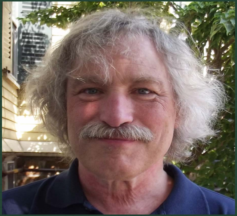
1. Introduction
2. Convolution Polynomial Rings
3. Multiplicative Inverse
4. `NTRUEncrypt`
5. Security
6. Performance
7. Conclusion

NTRU

- NTRU: *Nth-degree TRUncated polynomial ring* (pronounced *en-trū*)

NTRU

- `NTRU`: *Nth-degree TRUncated polynomial ring* (pronounced *en-trū*)
- A public key cryptosystem [HPS98] invented in early 1996 by



Hoffstein

Pipher

Silverman

Ring of Convolution Polynomials

Definition

The ring of convolution polynomials of rank N[1] is the quotient ring

$$R = \frac{\mathbb{Z}[x]}{\langle x^N - 1 \rangle}$$

[1] a.k.a. N-th truncated polynomial ring

Ring of Convolution Polynomials

Definition

The ring of convolution polynomials of rank N[1] is the quotient ring

$$R = \frac{\mathbb{Z}[x]}{\langle x^N - 1 \rangle}$$

Definition

The ring of convolution polynomials modulo q of rank N is the quotient ring

$$R_q = \frac{\mathbb{Z}_q[x]}{\langle x^N - 1 \rangle}$$

[1] a.k.a. N-th truncated polynomial ring

The Elements of Convolution Polynomial Rings

How does the elements of convolution polynomial rings look?

The Elements of Convolution Polynomial Rings

How does the elements of convolution polynomial rings look?

- Every element of R has a unique representation of the form

$$a_0 + a_1x + a_2x^2 + \cdots + a_{N-1}x^{N-1} = \sum_{i=0}^{N-1} a_i x^i \text{ or } \mathbf{a} = (a_0, \cdots, a_{N-1})$$

  with the coefficients in $\mathbb{Z}$.

The Elements of Convolution Polynomial Rings

How does the elements of convolution polynomial rings look?

- Every element of R has a unique representation of the form

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_{N-1} x^{N-1} = \sum_{i=0}^{N-1} a_i x^i \text{ or } \mathbf{a} = (a_0, \cdots, a_{N-1})$$

  with the coefficients in $\mathbb{Z}$.
- For every term x^k, if $k = r \mod N$, then

$$x^k = x^r.$$
$$x^N = 1, x^{N+1} = x, \ldots$$

The Elements of Convolution Polynomial Rings

How does the elements of convolution polynomial rings look?

- Every element of R has a unique representation of the form

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_{N-1} x^{N-1} = \sum_{i=0}^{N-1} a_i x^i \text{ or } \mathbf{a} = (a_0, \cdots, a_{N-1})$$

  with the coefficients in $\mathbb{Z}$.
- For every term x^k, if $k = r \mod N$, then

$$x^k = x^r.$$
$$x^N = 1, x^{N+1} = x, \ldots$$

- Polynomials in R_q can also be uniquely identified in the same way.

Operations of Convolution Polynomial Rings

Every ring has two operations, i.e, addition and multiplication.

- **Addition** of polynomials correspond to the usual addition of vectors,

$$a(x) + b(x) \leftrightarrow (a_0 + b_0, a_1 + b_1, a_2 + b_2, \dots, a_{N-1} + b_{N-1}).$$

Operations of Convolution Polynomial Rings

Every ring has two operations, i.e, addition and multiplication.

- **Addition** of polynomials correspond to the usual addition of vectors,

$$a(x) + b(x) \leftrightarrow (a_0 + b_0, a_1 + b_1, a_2 + b_2, \dots, a_{N-1} + b_{N-1}).$$

- **Multiply** two polynomials mod $x^N - 1$, i.e., replace x^k with $x^{k \mod N}$.

$$\mathbf{c} = \mathbf{a} \star \mathbf{b}, \quad c_i = \sum_{j=0}^{N-1} a_j b_{i-j}$$

Example

Let $N = 3$ and $a(x) = 1 + 3x + x^2$, and $b(x) = -4 + x + 2x^2$. Then

$$\begin{aligned} a(x) + b(x) &= (1-4) + (3+1)x + (1+2)x^2 = -3 + 4x + 3x^2 \\ a(x) \star b(x) &= -4 - 11x + x^2 + 7x^3 + 2x^4 \\ &= -4 - 11x + x^2 + 7 + 2x \\ &= 3 - 9x + x^2 \in R = \frac{\mathbb{Z}[x]}{\langle x^3 - 1 \rangle} \\ &= 3 + 5x + x^2 \in R_7 = \frac{\mathbb{Z}_7[x]}{\langle x^3 - 1 \rangle}. \end{aligned}$$

Example

Let $N = 3$ and $a(x) = 1 + 3x + x^2$, and $b(x) = -4 + x + 2x^2$. Then

$$\begin{aligned} a(x) + b(x) &= (1-4) + (3+1)x + (1+2)x^2 = -3 + 4x + 3x^2 \\ a(x) \star b(x) &= -4 - 11x + x^2 + 7x^3 + 2x^4 \\ &= -4 - 11x + x^2 + 7 + 2x \\ &= 3 - 9x + x^2 \in R = \frac{\mathbb{Z}[x]}{\langle x^3 - 1 \rangle} \\ &= 3 + 5x + x^2 \in R_7 = \frac{\mathbb{Z}_7[x]}{\langle x^3 - 1 \rangle}. \end{aligned}$$

$$\begin{aligned} \mathbf{a} \star \mathbf{b} = [a_0 \quad a_1 \quad a_2] \begin{bmatrix} b_0 & b_1 & b_2 \\ b_2 & b_0 & b_1 \\ b_1 & b_2 & b_0 \end{bmatrix} = [1 \quad 3 \quad 1] \begin{bmatrix} -4 & 1 & 2 \\ 2 & -4 & 1 \\ 1 & 2 & -4 \end{bmatrix} \\ = [3 \quad -9 \quad 1] \end{aligned}$$

Example

Let $N = 3$ and $a(x) = 1 + 3x + x^2$, and $b(x) = -4 + x + 2x^2$. Then

$$\begin{aligned}
a(x) + b(x) =& (1-4) + (3+1)x + (1+2)x^2 = -3 + 4x + 3x^2 \\
a(x) \star b(x) =& -4 - 11x + x^2 + 7x^3 + 2x^4 \\
=& -4 - 11x + x^2 + 7 + 2x \\
=& 3 - 9x + x^2 \in R = \frac{\mathbb{Z}[x]}{\langle x^3 - 1 \rangle} \\
=& 3 + 5x + x^2 \in R_7 = \frac{\mathbb{Z}_7[x]}{\langle x^3 - 1 \rangle}.
\end{aligned}$$

$$\mathbf{a} \star \mathbf{b} = [a_0 \quad a_1 \quad a_2] \begin{bmatrix} b_0 & b_1 & b_2 \\ b_2 & b_0 & b_1 \\ b_1 & b_2 & b_0 \end{bmatrix} = [1 \quad 3 \quad 1] \begin{bmatrix} -4 & 1 & 2 \\ 2 & -4 & 1 \\ 1 & 2 & -4 \end{bmatrix}$$
$$= [3 \quad -9 \quad 1]$$

A polynomial multiplication takes N^2 multiplications.

Convolution Polynomial Rings in Sage I

- Generate $R = \frac{\mathbb{Z}[x]}{\langle x^7 - 1 \rangle}$:

```
N = 7
ZX.<X> = PolynomialRing(ZZ)
R.<x> = ZX.quotient(X^N - 1); R
Univariate Quotient Polynomial Ring in x over
Integer Ring with modulus X^7 - 1
```

- Generate $R_3 = \frac{\mathbb{Z}_3[x]}{\langle x^7 - 1 \rangle}$

```
N, q = 7, 3
ZqX.<X> = PolynomialRing(Zmod(q))
Rq.<x> = ZqX.quotient(X^N - 1); Rq
Univariate Quotient Polynomial Ring in x over
Ring of integers modulo 3 with modulus X^7 + 2
```

Convolution Polynomial Rings in Sage II

- Choose two elements at random from R, and multiply them:

```
[f, g] = [Rq.random_element() for _ in range(2)]
print("(f, g) = ", (f, g))
print("f*g = ", f*g)
(f, g) =  (2*x^6 + 2*x^4 + x^3, 2*x^6 + x^2 + 2*x)
f*g =  2*x^6 + 2*x^4 + x^3 + 2*x^2 + 2*x + 1
```

- Lift $f \in R_3 = \frac{\mathbb{Z}_3[X]}{\langle X^7 - 1 \rangle}$ into $\mathbb{Z}_3[X]$

```
print(f.parent())
Univariate Quotient Polynomial Ring in x over
Ring of integers modulo 3 with modulus X^7 + 2

f = f.lift()
print(f.parent())
Univariate Polynomial Ring in X over
Ring of integers modulo 3
```

Multiplicative Inverse I

$f(x) \in R_q$ has a multiplicative inverse if and only if

$$\gcd(f(x), x^N - 1) = 1 \in \mathbb{Z}_q[x].$$

If so, then the inverse $f(x)^{-1} \in R_q$ can be computed using the extended Euclidean algorithm to find polynomials $u(x), v(x) \in \mathbb{Z}_q[x]$ satisfying

$$f(x) \star u(x) + (x^N - 1) \star v(x) = 1.$$

Then $f^{-1}(x) = u(x) \in R_q$.

Multiplicative Inverse II

- You can simply compute the inverse via SageMath[The21] (if it exists!)

```
reset()
N, q = 7, 4
Zx.<X> = ZZ[]
f = X^6 - X^4 + X^3 + X^2 -1
Zq.<a> = PolynomialRing(Zmod(q))
f = Zq(f) # Moving f from Zx[x] into Zq[a]
print("gcd(f, a^N - 1) = ", f.gcd(a^N - 1))
f_inv = f.inverse_mod(a^N - 1); f_inv(a=X)

gcd(f, a^N - 1) =  1
X^5 + 3*X^4 + 3*X^3 + 2*X^2
```

- Check to see if the multiplication of $f \star f^{-1} = 1 \mod q$?

```
Zq(f*f_inv).mod(a^N - 1)

1
```

NTRUEncrypt

Parameters: $N, p, q, (p, q) = 1$. E.g., $N = 401, p = 3, q = 2048$

Parameters: $N, p, q,\ (p,q)=1$. E.g., $N=401, p=3, q=2048$

Definition (Centered modular reduction)

For an odd integer n and integers a and b, define

$$a \bmod n = b \text{ if } a \equiv b \bmod n \text{ and } -\frac{n-1}{2} \le b \le \frac{n}{2}.$$

For example $a \bmod 5 \in \{-2,-1,0,1,2\}$, whereas a $\bmod 5 \in \{0,1,2,3,4\}$.

NTRUEncrypt

- Key-Generation:
  - Choose $F(x), G(x) \in R$ s.t. $\mathbf{F}, \mathbf{G} \in \{-1, 0, 1\}^N$.

NTRUEncrypt

- Key-Generation:
  - Choose $F(x), G(x) \in R$ s.t. $\mathbf{F}, \mathbf{G} \in \{-1, 0, 1\}^N$.
  - $f(x) = 1 + pF(x)$, compute $f^{-1}(x)$
  - $g(x) = pG(x)$
  - Compute $h(x) = f^{-1}(x) \star g(x) \bmod q$

NTRUEncrypt

- Key-Generation:
  - Choose $F(x), G(x) \in R$ s.t. $\mathbf{F}, \mathbf{G} \in \{-1, 0, 1\}^N$.
  - $f(x) = 1 + pF(x)$, compute $f^{-1}(x)$
  - $g(x) = pG(x)$
  - Compute $h(x) = f^{-1}(x) \star g(x) \bmod q$
  - PK: $h(x)$, SK: $f(x)$

NTRUEncrypt

- Key-Generation:
  - Choose $F(x), G(x) \in R$ s.t. $\mathbf{F}, \mathbf{G} \in \{-1, 0, 1\}^N$.
  - $f(x) = 1 + pF(x)$, compute $f^{-1}(x)$
  - $g(x) = pG(x)$
  - Compute $h(x) = f^{-1}(x) \star g(x) \bmod q$
  - PK: $h(x)$, SK: $f(x)$
- Enc:
  - Plaintext $\mathbf{m} \in \{-1, 0, 1\}^N$
  - Choose $\mathbf{r} \in \{-1, 0, 1\}^N$ uniformly at random
  - Ciphertext $\mathbf{y} = \mathbf{r} \star \mathbf{h} + \mathbf{m} \bmod q$

NTRUEncrypt

- Key-Generation:
  - Choose $F(x), G(x) \in R$ s.t. $\mathbf{F}, \mathbf{G} \in \{-1, 0, 1\}^N$.
  - $f(x) = 1 + pF(x)$, compute $f^{-1}(x)$
  - $g(x) = pG(x)$
  - Compute $h(x) = f^{-1}(x) \star g(x) \bmod q$
  - PK: $h(x)$, SK: $f(x)$
- Enc:
  - Plaintext $\mathbf{m} \in \{-1, 0, 1\}^N$
  - Choose $\mathbf{r} \in \{-1, 0, 1\}^N$ uniformly at random
  - Ciphertext $\mathbf{y} = \mathbf{r} \star \mathbf{h} + \mathbf{m} \bmod q$
- Dec:
  - Compute $\mathbf{a} = \mathbf{f} \star \mathbf{y} \bmod q$
  - Compute $\mathbf{m}' = \mathbf{a} \bmod p$

How does the decryption work?

How does the decryption work?

- Dec:
  - Compute $\mathbf{a} = \mathbf{f} \star \mathbf{y} \bmod q$

$$\begin{aligned}\mathbf{a} &= \mathbf{f} \star \mathbf{r} \star \mathbf{h} + \mathbf{f} \star \mathbf{m} \bmod q \quad (\mathbf{y} = \mathbf{r} \star \mathbf{h} + \mathbf{m} \bmod q) \\ &= \mathbf{f} \star \mathbf{r} \star \mathbf{f}^{-1} \star \mathbf{g} + \mathbf{f} \star \mathbf{m} \bmod q \quad (\mathbf{h} = \mathbf{f}^{-1} \star \mathbf{g} \bmod q) \\ &\equiv \mathbf{r} \star \mathbf{g} + \mathbf{f} \star \mathbf{m} \bmod q\end{aligned}$$

How does the decryption work?

- Dec:
  - Compute $\mathbf{a} = \mathbf{f} \star \mathbf{y} \bmod q$

$$\begin{aligned} \mathbf{a} &= \mathbf{f} \star \mathbf{r} \star \mathbf{h} + \mathbf{f} \star \mathbf{m} \bmod q \quad (\mathbf{y} = \mathbf{r} \star \mathbf{h} + \mathbf{m} \bmod q) \\ &= \mathbf{f} \star \mathbf{r} \star \mathbf{f}^{-1} \star \mathbf{g} + \mathbf{f} \star \mathbf{m} \bmod q \quad (\mathbf{h} = \mathbf{f}^{-1} \star \mathbf{g} \bmod q) \\ &\equiv \mathbf{r} \star \mathbf{g} + \mathbf{f} \star \mathbf{m} \bmod q \end{aligned}$$

  - Compute $\mathbf{m}' = \mathbf{a} \bmod p$
    If the coefficients of $\mathbf{r} \star \mathbf{g} + \mathbf{f} \star \mathbf{m}$ lie in the interval

$$\left[-\frac{q-1}{2}, \frac{q}{2} \right],$$

    which holds with high probability. In such cases,

$$\mathbf{a} = \mathbf{f} \star \mathbf{y} \bmod q = \mathbf{r} \star \mathbf{g} + \mathbf{f} \star \mathbf{m}$$

$$\begin{aligned} \mathbf{m}' = (\mathbf{f} \star \mathbf{y} \bmod q) \quad \bmod p &= \mathbf{r} \star \mathbf{g} + \mathbf{f} \star \mathbf{m} \bmod p \\ &\equiv \mathbf{m} \bmod p. \quad (\mathbf{g} = \mathbf{pG}, \mathbf{f} = \mathbf{1} + \mathbf{pF}) \end{aligned}$$

How does the decryption work?

- Dec:
  - Compute $\mathbf{a} = \mathbf{f} \star \mathbf{y} \bmod q$

$$\begin{aligned} \mathbf{a} &= \mathbf{f} \star \mathbf{r} \star \mathbf{h} + \mathbf{f} \star \mathbf{m} \bmod q \quad (\mathbf{y} = \mathbf{r} \star \mathbf{h} + \mathbf{m} \bmod q) \\ &= \mathbf{f} \star \mathbf{r} \star \mathbf{f}^{-1} \star \mathbf{g} + \mathbf{f} \star \mathbf{m} \bmod q \quad (\mathbf{h} = \mathbf{f}^{-1} \star \mathbf{g} \bmod q) \\ &\equiv \mathbf{r} \star \mathbf{g} + \mathbf{f} \star \mathbf{m} \bmod q \end{aligned}$$

  - Compute $\mathbf{m}' = \mathbf{a} \bmod p$
    If the coefficients of $\mathbf{r} \star \mathbf{g} + \mathbf{f} \star \mathbf{m}$ lie in the interval

$$\left[-\frac{q-1}{2}, \frac{q}{2}\right],$$

    which holds with high probability. In such cases,

$$\mathbf{a} = \mathbf{f} \star \mathbf{y} \bmod q = \mathbf{r} \star \mathbf{g} + \mathbf{f} \star \mathbf{m}$$

$$\begin{aligned} \mathbf{m}' = (\mathbf{f} \star \mathbf{y} \bmod q) \quad \bmod p &= \mathbf{r} \star \mathbf{g} + \mathbf{f} \star \mathbf{m} \bmod p \\ &\equiv \mathbf{m} \bmod p. \quad (\mathbf{g} = \mathbf{pG}, \mathbf{f} = \mathbf{1} + \mathbf{pF}) \end{aligned}$$

    Therefore, $\mathbf{m}' = \mathbf{m} =$. The ciphertext is decrypted correctly.

How does the decryption work?

If an attacker decrypts $\mathbf{y}$ with a $\mathbf{f}'$ where $\mathbf{f}' \neq \mathbf{f}$, can she/he recover the plaintext polynomial $\mathbf{m}$?

Decrypt with f'

Decrypt with $\mathbf{f}'$

- Dec:
  - Compute $\mathbf{a} = \mathbf{f}' \star \mathbf{y} \bmod q$

$$\mathbf{a} = \mathbf{f}' \star \mathbf{r} \star \mathbf{h} + \mathbf{f}' \star \mathbf{m} \bmod q \quad (\mathbf{y} = \mathbf{r} \star \mathbf{h} + \mathbf{m} \bmod q)$$

$$= \mathbf{f}' \star \mathbf{r} \star \mathbf{f}^{-1} \star \mathbf{g} + \mathbf{f}' \star \mathbf{m} \bmod q \quad (\mathbf{h} = \mathbf{f}^{-1} \star \mathbf{g} \bmod q)$$

$$\equiv \mathbf{r} \star \mathbf{g} + \mathbf{f} \star \mathbf{m} \bmod q$$

Decrypt with $\mathbf{f}'$

- Dec:
  - Compute $\mathbf{a} = \mathbf{f}' \star \mathbf{y} \bmod q$

$$\mathbf{a} = \mathbf{f}' \star \mathbf{r} \star \mathbf{h} + \mathbf{f}' \star \mathbf{m} \bmod q \quad (\mathbf{y} = \mathbf{r} \star \mathbf{h} + \mathbf{m} \bmod q)$$
$$= \mathbf{f}' \star \mathbf{r} \star \mathbf{f}^{-1} \star \mathbf{g} + \mathbf{f}' \star \mathbf{m} \bmod q \quad (\mathbf{h} = \mathbf{f}^{-1} \star \mathbf{g} \bmod q)$$
$$\equiv \mathbf{r} \star \mathbf{g} + \mathbf{f} \star \mathbf{m} \bmod q$$

  - Compute $\mathbf{m}' = \mathbf{a} \bmod p$

    If the following equation holds, the attacker can recover $\mathbf{m}$.

$$\mathbf{a} = \mathbf{f}' \star \mathbf{y} \bmod q = \mathbf{f}'' \star \mathbf{r} \star \mathbf{g} + \mathbf{f}' \star \mathbf{m}$$

where $\mathbf{f}'' = \mathbf{f}' \star \mathbf{f}^{-1} \bmod q$ Recall $\mathbf{g} = \mathbf{pG}, \mathbf{f} = \mathbf{1} + \mathbf{pF}$

$\mathbf{f}' \star \mathbf{m}$ and $\mathbf{r} \star \mathbf{g}$ still have small coefficients, whereas $\mathbf{f}'' \star \mathbf{r} \star \mathbf{g}$ is likely to have large coefficients.

Example I

Suppose $N = 11, p = 3$ and $q = 23$.

Key-Generation:

- Choose $F(x), G(x) \in R$ s.t. $\mathbf{F}, \mathbf{G} \in \{-1, 0, 1\}^N$.
  $F(x) = x^{10} - x^9 + x^8 - x^4 - x^2 + x$
  $f(x) = 3x^{10} - 3x^9 + 3x^8 - 3x^4 - 3x^2 + 3x + 1 \leftarrow f(x) = 1 + pF(x)$
  $f^{-1}(x) = -11x^{10} + 7x^9 - 8x^8 + 2x^7 + 6x^6 - x^5 - 2x^4 - 3x^3 - 3x^2 - 11x + 2$
  $G(x) = x^9 - x^8 - x^7 + x^6 + x^4 - 1,$
  $g(x) = 3x^9 - 3x^8 - 3x^7 + 3x^6 + 3x^4 - 3 \leftarrow g(x) = pG(x)$
- Compute $h(x) = f^{-1}(x) \star g(x) \bmod q$
  $h(x) = 7x^{10} - 8x^9 + 3x^8 - 10x^6 - 8x^5 - 6x^3 - 8x^2 + 4x + 3$
- PK: $h(x)$, SK: $f(x)$

Example II

Enc:

- Plaintext $\mathbf{m} \in \{-1, 0, 1\}^N$
  $m(x) = x^{10} - x^5 + x^3 - 1$
- Choose $\mathbf{r} \in \{-1, 0, 1\}^N$ uniformly at random
  $r(x) = x^9 + x^7 - x^6 - x^5 - x^4 + x^2.$
- Ciphertext $\mathbf{y} = \mathbf{r} \star \mathbf{h} + \mathbf{m} \bmod q$
  $y(x) = -3x^{10} + 9x^9 + -8x^8 - 3x^7 + 11x^6 - 6x^5 + 6x^4 - 5x^3 - 2x^2 + 1$

Example III

Dec:

- Compute $\mathbf{a} = \mathbf{f} \star \mathbf{y} \bmod q = \mathbf{f} \star \mathbf{r} \star \mathbf{f}^{-1} \star \mathbf{g} + \mathbf{f} \star \mathbf{m} \bmod q$
  $\mathbf{f} \star \mathbf{y} = 12x^{10} - 29x^8 + 3x^7 + 23x^6 + 45x^5 - 66x^4 + 67x^3 - 83x^2 + 63x - 35 \in R$
  $a(x) = -11x^{10} - 6x^8 + 3x^7 - x^5 + 3x^4 - 2x^3 + 9x^2 - 6x + 11 \in R_q$
- Compute $\mathbf{m}' = \mathbf{a} \bmod p$
  Coefficients of $a(x)$ all lie in the interval $[-11, 11]$. Applying mod 3 we have

$$m'(x) = x^{10} - x^5 + x^3 - 1 = m(x).$$

- Check
  $\mathbf{r} \star \mathbf{g} + \mathbf{f} \star \mathbf{m} = -11x^{10} - 6x^8 + 3x^7 - x^5 + 3x^4 - 2x^3 + 9x^2 - 6x + 11 \in R$

Example IV

Dec with incorect secret key $\mathbf{f}' = 1 + 3(x^9 - x^8 - x^6 - x^5 + x^3 + 1)$

- Compute $\mathbf{a} = \mathbf{f}' \star \mathbf{y} \bmod q = \mathbf{f}' \star \mathbf{r} \star \mathbf{f}^{-1} \star \mathbf{g} + \mathbf{f}' \star \mathbf{m} \bmod q$
  $a(x) = 8x^{10} + 5x^8 + 2x^7 - 2x^6 - 9x^5 + 2x^4 - 2x^3 + 6x^2 - 7x - 3 \in R_q$
- Compute $\mathbf{m}' = \mathbf{a} \bmod p$
  $m'(x) = -x^{10} - x^8 - x^7 + x^6 - x^4 + x^3 - x \neq m(x).$
- Check
  $\mathbf{f}'' = \mathbf{f}' \star \mathbf{f}^{-1} \bmod q = -7x^{10} - 9x^9 + 4x^8 - 4x^7 + 6x^6 - 7x^5 - 3x^4 + 3x^3 + 2x^2 - 11x + 4 \in R_q$
  $\mathbf{f}' \star \mathbf{m} = 7x^{10} - 6x^9 - 3x^7 + 6x^6 - 4x^5 - 3x^4 - 2x^3 + 6x^2 + 3x - 4 \in R$
  $\mathbf{r} \star \mathbf{g} = -9x^{10} - 9x^9 + 3x^6 + 3x^5 - 3x^3 + 6x^2 + 3x + 6$
  $\mathbf{f}' \star \mathbf{f}^{-1} \star \mathbf{r} \star \mathbf{g} = 24x^{10} + 213x^9 - 87x^8 - 18x^7 + 15x^6 - 51x^5 + 51x^4 - 69x^3 - 138x^2 - 33x + 93$
  $\mathbf{f}'' \star \mathbf{r} \star \mathbf{g}$ has large coefficients compared to $q/2$.

Conditions for parameters

- Each of $\mathbf{F}, \mathbf{G}, \mathbf{r}, \mathbf{m}$ have (roughly) $\frac{1}{3}$ of their coefficients equal to each of $-1, 0$ and 1.
  - Related to the security of the scheme.
- q should be large compared to N.
  - To ensure the decryption is correct with high probability.

What is the hard math problem behind NTRU?

- **Lattice reduction**
  - Same problem that breaks the knapsack!
- If attacker can determine $\mathbf{f}^{-1}$ or $\mathbf{g}$, from $\mathbf{h}$, she gets the private key.

What is the hard math problem behind NTRU?

- **Lattice reduction**
  - Same problem that breaks the knapsack!
- If attacker can determine $\mathbf{f}^{-1}$ or $\mathbf{g}$, from $\mathbf{h}$, she gets the private key.

The NTRU Key Recovery Problem[HPSS08]

Given $h(x)$, find **ternary** polynomials $f(x)$ and $g(x)$ satisfying $f(x) \star h(x) = g(x) \mod q$ where coefficients of $f(x)$ and $g(x)$ lie in $\{-p, 0, p\}$.

What is the hard math problem behind NTRU?

- Recall $\mathbf{h} = \mathbf{f}^{-1} \star \mathbf{g} \mod q$

What is the hard math problem behind NTRU?

- Recall $\mathbf{h} = \mathbf{f}^{-1} \star \mathbf{g} \mod q$
- Equivalently, $\mathbf{f} \star \mathbf{h} \equiv \mathbf{g} \mod q$. I.e., there exists some integer vector $\mathbf{t}$ such that

$$\mathbf{f} \star \mathbf{h} - \mathbf{g} = q\mathbf{t}$$

What is the hard math problem behind NTRU?

- Recall $\mathbf{h} = \mathbf{f}^{-1} \star \mathbf{g} \mod q$
- Equivalently, $\mathbf{f} \star \mathbf{h} \equiv \mathbf{g} \mod q$. I.e., there exists some integer vector $\mathbf{t}$ such that

$$\mathbf{f} \star \mathbf{h} - \mathbf{g} = q\mathbf{t}$$

- Let

$$\mathbf{H} = \begin{pmatrix} h_0 & h_{N-1} & h_{N-2} & \cdots & h_1 \\ h_1 & h_0 & h_{N-1} & \cdots & h_2 \\ \vdots & & \ddots & & \vdots \\ h_{N-1} & h_{N-2} & h_{N-3} & \cdots & h_0 \end{pmatrix}, \quad \mathbf{M} = \begin{pmatrix} \mathbf{I}_{N\times N} & \mathbf{H}_{N\times N} \\ \mathbf{0}_{N\times N} & q\mathbf{I}_{N\times N} \end{pmatrix}$$

So $(\mathbf{f}, -\mathbf{t})\mathbf{M} = (\mathbf{f}, \mathbf{g})$.

What is the hard math problem behind NTRU?

- Recall $\mathbf{h} = \mathbf{f}^{-1} \star \mathbf{g} \mod q$
- Equivalently, $\mathbf{f} \star \mathbf{h} \equiv \mathbf{g} \mod q$. I.e., there exists some integer vector $\mathbf{t}$ such that

$$\mathbf{f} \star \mathbf{h} - \mathbf{g} = q\mathbf{t}$$

- Let

$$\mathbf{H} = \begin{pmatrix} h_0 & h_{N-1} & h_{N-2} & \cdots & h_1 \\ h_1 & h_0 & h_{N-1} & \cdots & h_2 \\ \vdots & & \ddots & & \vdots \\ h_{N-1} & h_{N-2} & h_{N-3} & \cdots & h_0 \end{pmatrix}, \quad \mathbf{M} = \begin{pmatrix} \mathbf{I}_{N\times N} & \mathbf{H}_{N\times N} \\ \mathbf{0}_{N\times N} & q\mathbf{I}_{N\times N} \end{pmatrix}$$

  So $(\mathbf{f}, -\mathbf{t})\mathbf{M} = (\mathbf{f}, \mathbf{g})$.
- Let $\mathcal{L}$ be the lattice spanned by column vectors of $\mathbf{M}$.

What is the hard math problem behind NTRU?

- Recall $\mathbf{h} = \mathbf{f}^{-1} \star \mathbf{g} \mod q$
- Equivalently, $\mathbf{f} \star \mathbf{h} \equiv \mathbf{g} \mod q$. I.e., there exists some integer vector $\mathbf{t}$ such that

$$\mathbf{f} \star \mathbf{h} - \mathbf{g} = q\mathbf{t}$$

- Let

$$\mathbf{H} = \begin{pmatrix} h_0 & h_{N-1} & h_{N-2} & \cdots & h_1 \\ h_1 & h_0 & h_{N-1} & \cdots & h_2 \\ \vdots & & \ddots & & \vdots \\ h_{N-1} & h_{N-2} & h_{N-3} & \cdots & h_0 \end{pmatrix}, \quad \mathbf{M} = \begin{pmatrix} \mathbf{I}_{N\times N} & \mathbf{H}_{N\times N} \\ \mathbf{0}_{N\times N} & q\mathbf{I}_{N\times N} \end{pmatrix}$$

So $(\mathbf{f}, -\mathbf{t})\mathbf{M} = (\mathbf{f}, \mathbf{g})$.

- Let $\mathcal{L}$ be the lattice spanned by column vectors of $\mathbf{M}$. Then

$$(\mathbf{f}, \mathbf{g}) \in \mathcal{L}$$

.

What is the hard math problem behind NTRU?

The norm of vector $(\mathbf{f}, \mathbf{g})$

- Each of $\mathbf{F}, \mathbf{G}$ have (roughly) $\frac{1}{3}$ of their coefficients equal to each of $-1, 0$ and 1.

What is the hard math problem behind NTRU?

The norm of vector $(\mathbf{f}, \mathbf{g})$

- Each of $\mathbf{F}, \mathbf{G}$ have (roughly) $\frac{1}{3}$ of their coefficients equal to each of $-1, 0$ and 1.
- Each of $\mathbf{f}, \mathbf{g}$ have (roughly) $\frac{1}{3}$ of their coefficients equal to each of $-p, 0$ and p. $\leftarrow f(x) = 1 + pF(x), g(x) = pG(x)$

What is the hard math problem behind NTRU?

The norm of vector $(\mathbf{f}, \mathbf{g})$

- Each of $\mathbf{F}, \mathbf{G}$ have (roughly) $\frac{1}{3}$ of their coefficients equal to each of $-1, 0$ and 1.
- Each of $\mathbf{f}, \mathbf{g}$ have (roughly) $\frac{1}{3}$ of their coefficients equal to each of $-p, 0$ and p. $\leftarrow f(x) = 1 + pF(x), g(x) = pG(x)$
- The norm of $(\mathbf{f}, \mathbf{g})$ is approximately

$$\sqrt{4Np^2/3} = 2\sqrt{3N} \text{ when } p = 3.$$

What is the hard math problem behind NTRU?

The norm of vector $(\mathbf{f}, \mathbf{g})$

- Each of $\mathbf{F}, \mathbf{G}$ have (roughly) $\frac{1}{3}$ of their coefficients equal to each of $-1, 0$ and 1.
- Each of $\mathbf{f}, \mathbf{g}$ have (roughly) $\frac{1}{3}$ of their coefficients equal to each of $-p, 0$ and p. $\leftarrow f(x) = 1 + pF(x), g(x) = pG(x)$
- The norm of $(\mathbf{f}, \mathbf{g})$ is approximately

$$\sqrt{4Np^2/3} = 2\sqrt{3N} \text{ when } p = 3.$$

However, a vector of length $2N$ whose coordinates take on random values in $[-q/2, q/2]$ would have norm approximately equal to

$$q\sqrt{N/6},$$

which is much larger (recall $q = 2048$).

What is the hard math problem behind NTRU?

The norm of vector $(\mathbf{f}, \mathbf{g})$

- Each of $\mathbf{F}, \mathbf{G}$ have (roughly) $\frac{1}{3}$ of their coefficients equal to each of $-1, 0$ and 1.
- Each of $\mathbf{f}, \mathbf{g}$ have (roughly) $\frac{1}{3}$ of their coefficients equal to each of $-p, 0$ and p. $\leftarrow f(x) = 1 + pF(x), g(x) = pG(x)$
- The norm of $(\mathbf{f}, \mathbf{g})$ is approximately

$$\sqrt{4Np^2/3} = 2\sqrt{3N} \text{ when } p = 3.$$

However, a vector of length $2N$ whose coordinates take on random values in $[-q/2, q/2]$ would have norm approximately equal to

$$q\sqrt{N/6},$$

which is much larger (recall $q = 2048$).

- It seems that $(\mathbf{f}, \mathbf{g})$ is the shortest vector in the lattice $\mathcal{L}$.

NTRU and SVP

- There is no proof that breaking `NTRUEncrypt` is as hard as solving the **Shortest Vector Problem** or the **Closest Vector Problem**.

NTRU and SVP

- There is no proof that breaking `NTRUEncrypt` is as hard as solving the **Shortest Vector Problem** or the **Closest Vector Problem**.
- In 2013, Damien Stehle and Ron Steinfeld created a provably secure version of NTRU [SS13].
- The European Union's PQCRYPTO project (Horizon 2020 ICT-645622) is evaluating the provably secure Stehle–Steinfeld version of NTRU as a potential European standard. However, the Stehle-Steinfeld version of NTRU is "significantly less efficient than the original scheme."

How Fast is NTRUEncrypt?

- The most time consuming part of encryption and decryption is the polynomial multiplication

How Fast is `NTRUEncrypt`?

- The most time consuming part of encryption and decryption is the polynomial multiplication
- Each coefficient is essentially the dot product of two vectors

How Fast is `NTRUEncrypt`?

- The most time consuming part of encryption and decryption is the polynomial multiplication
- Each coefficient is essentially the dot product of two vectors
- A polynomial multiplication of two polynomial of length N requires N^2 multiplications

How Fast is `NTRUEncrypt`?

- The most time consuming part of encryption and decryption is the polynomial multiplication
- Each coefficient is essentially the dot product of two vectors
- A polynomial multiplication of two polynomial of length N requires N^2 multiplications
- Hence, both encryption and decryption take $\mathcal{O}(N^2)$ steps, where each step is extremely fast.

How Fast is NTRUEncrypt?

- The most time consuming part of encryption and decryption is the polynomial multiplication
- Each coefficient is essentially the dot product of two vectors
- A polynomial multiplication of two polynomial of length N requires N^2 multiplications
- Hence, both encryption and decryption take $\mathcal{O}(N^2)$ steps, where each step is extremely fast.
- Faster than RSA at equivalent cryptographic strength.

How Fast is `NTRUEncrypt`?

- The most time consuming part of encryption and decryption is the polynomial multiplication
- Each coefficient is essentially the dot product of two vectors
- A polynomial multiplication of two polynomial of length N requires N^2 multiplications
- Hence, both encryption and decryption take $\mathcal{O}(N^2)$ steps, where each step is extremely fast.
- Faster than RSA at equivalent cryptographic strength.
- Promising PQC candidate

The National Institute of Standards and Technology wrote in a 2009 survey that "[there] are viable alternatives for both public key encryption and signatures that are not vulnerable to Shor's Algorithm" and "[of] the various lattice based cryptographic schemes that have been developed, the NTRU family of cryptographic algorithms appears to be the most practical".

Conclusion

- A lattice-based public key cryptosystem
- Its security relies on difficulty of `SVP` problem
- Has evolved since its introduction
- Considered theoretically sound
- Unlike `RSA` and `ECC`, `NTRU` is **not** known to be vulnerable against quantum computer based attack
- It has been standardized (IEEE Std 1363.1, X9.98)

Thanks for your attention!

Question?

References I

- Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman, *Ntru: A ring-based public key cryptosystem*, International Algorithmic Number Theory Symposium, Springer, 1998, pp. 267–288.
- Jeffrey Hoffstein, Jill Pipher, Joseph H Silverman, and Joseph H Silverman, *An introduction to mathematical cryptography*, vol. 1, Springer, 2008.
- Damien Stehlé and Ron Steinfeld, *Making ntruencrypt and ntrusign as secure as standard worst-case problems over ideal lattices*, IACR Cryptol. ePrint Arch. **2013** (2013), 4.
- The Sage Developers, *Sagemath, the Sage Mathematics Software System (Version 9.2.0)*, 2021, `https://www.sagemath.org`.