NTRU
Lattice-based PKCS

Logan Collins

April 21, 2015

What Do These Words Mean?

NTRU

NTRU is officially short for "N-th degree *TRU*ncated polynomial ring". However, it is also colloquially called "Number Theorists aRe Us".

PKCS

PKCS stands for Public Key CryptoSystem. These are asymmetric cryptographic schemes where the public key is a published value associated with a user for encryption and where private key is used for decryption for the associated user.

What Do These Words Mean? (cont.)

Lattice

In Cryptography, a lattice is a vector space generated with linear combinations of its basis with only integer coefficients.

Super Quick Reminder

Super Quick Reminder

Alice chooses a large prime q, which is public.

Super Quick Reminder

Alice chooses a large prime q, which is public. Then, Alice chooses f and g satisfying $f < \sqrt{q/2}$ and $\sqrt{q/4} < g < \sqrt{q/2}$, with f and g coprime.

Super Quick Reminder

Alice chooses a large prime q, which is public. Then, Alice chooses f and g satisfying $f < \sqrt{q/2}$ and $\sqrt{q/4} < g < \sqrt{q/2}$, with f and g coprime. Alice computes $h \equiv f^{-1}g \bmod q$ which completes her public key.

Super Quick Reminder

Alice chooses a large prime q, which is public. Then, Alice chooses f and g satisfying $f < \sqrt{q/2}$ and $\sqrt{q/4} < g < \sqrt{q/2}$, with f and g coprime. Alice computes $h \equiv f^{-1}g \bmod q$ which completes her public key. For Bob to send a message to Alice, he sends Alice $c \equiv rh + m \bmod q$ with $m < \sqrt{q/4}$ and $r < \sqrt{q/2}$.

Super Quick Reminder

Alice chooses a large prime q, which is public. Then, Alice chooses f and g satisfying $f < \sqrt{q/2}$ and $\sqrt{q/4} < g < \sqrt{q/2}$, with f and g coprime. Alice computes $h \equiv f^{-1}g \bmod q$ which completes her public key. For Bob to send a message to Alice, he sends Alice $c \equiv rh + m \bmod q$ with $m < \sqrt{q/4}$ and $r < \sqrt{q/2}$. To decrypt, Alice computes $a \equiv fc \bmod q$ and then $b \equiv f^{-1}a \bmod g$.

Super Quick Reminder

Alice chooses a large prime q, which is public. Then, Alice chooses f and g satisfying $f < \sqrt{q/2}$ and $\sqrt{q/4} < g < \sqrt{q/2}$, with f and g coprime. Alice computes $h \equiv f^{-1}g \bmod q$ which completes her public key. For Bob to send a message to Alice, he sends Alice $c \equiv rh + m \bmod q$ with $m < \sqrt{q/4}$ and $r < \sqrt{q/2}$. To decrypt, Alice computes $a \equiv fc \bmod q$ and then $b \equiv f^{-1}a \bmod g$. We claim that $b = m$.

Super Quick Reminder

Alice chooses a large prime q, which is public. Then, Alice chooses f and g satisfying $f < \sqrt{q/2}$ and $\sqrt{q/4} < g < \sqrt{q/2}$, with f and g coprime. Alice computes $h \equiv f^{-1}g \bmod q$ which completes her public key. For Bob to send a message to Alice, he sends Alice $c \equiv rh + m \bmod q$ with $m < \sqrt{q/4}$ and $r < \sqrt{q/2}$. To decrypt, Alice computes $a \equiv fc \bmod q$ and then $b \equiv f^{-1}a \bmod g$. We claim that $b = m$. Notice $a \equiv fc \equiv f(rh + m) \equiv frf^{-1}g + fm \equiv rg + fm \bmod q$. Then we compute $b \equiv f^{-1}(rg + fm) \bmod g \equiv m \bmod g$.

Attacks on this System

Attacks on this System

The private key in the system is (f, g) and we can verify that two numbers work by computing $f^{-1}g == h$. However, for large primes, this is an impractically large space.

Attacks on this System

The private key in the system is (f, g) and we can verify that two numbers work by computing $f^{-1}g == h$. However, for large primes, this is an impractically large space.

Note that we can rewrite this attack as $Fh \equiv G \bmod q \to Fh = G + qR$ and rewrite this as $F(1, h) - R(0, q) = (F, G)$.

Attacks on this System

The private key in the system is (f, g) and we can verify that two numbers work by computing $f^{-1}g == h$. However, for large primes, this is an impractically large space.

Note that we can rewrite this attack as $Fh \equiv G \bmod q \rightarrow Fh = G + qR$ and rewrite this as $F(1, h) - R(0, q) = (F, G)$.

That is, we are trying to find a short vector in a lattice with basis $\{(1, h), (0, q)\}$, where (h, q) is Alice's public key.

Attacks on this System (cont.)

Unfortunately, our favorite mathematician, Gauss, figured out how to rapidly solve such systems a long, long time ago.

Attacks on this System (cont.)

Unfortunately, our favorite mathematician, Gauss, figured out how to rapidly solve such systems a long, long time ago.

To find these short vectors, we first label our basis $\mathbf{v_1}, \mathbf{v_2}$ with $||\mathbf{v_2}|| > ||\mathbf{v_1}||$, swapping if necessary. We compute $m = \left\lfloor \frac{\mathbf{v_1}\mathbf{v_2}}{||\mathbf{v_1}||^2} \right\rceil$. If $m = 0$, we return $\{\mathbf{v_1}, \mathbf{v_2}\}$ where $\mathbf{v_1}$ is (provably) the shortest vector in the lattice. Otherwise, $\mathbf{v_2} = \mathbf{v_2} - m\mathbf{v_1}$ and we continue.

Now What?

Now What?

Gauss's Lattice Reduction algorithm works well in $\mathbb{R}^2$, but not well in higher dimensions... so...

Now What?

Gauss's Lattice Reduction algorithm works well in $\mathbb{R}^2$, but not well in higher dimensions... so...

Add more dimensions!

NTRU

Remember that NTRU stands for the Nth degree truncated polynomial ring?

NTRU

Remember that NTRU stands for the Nth degree truncated polynomial ring?

Well, that is because NTRU operates in $\frac{\mathbb{Z}[x]}{x^N-1}$, $\frac{\mathbb{Z}_q[x]}{x^N-1}$, and $\frac{\mathbb{Z}_p[x]}{x^N-1}$.

NTRU

Remember that NTRU stands for the Nth degree truncated polynomial ring?

Well, that is because NTRU operates in $\frac{\mathbb{Z}[x]}{x^N-1}$, $\frac{\mathbb{Z}_q[x]}{x^N-1}$, and $\frac{\mathbb{Z}_p[x]}{x^N-1}$. This forms a lattice in $\mathbb{R}^{2N}$!

Convolution Polynomial Rings

That is, NTRU operates on convolution polynomial rings $R = \frac{\mathbb{Z}[x]}{x^N - 1}$, $R_q = \frac{\mathbb{Z}_q[x]}{x^N - 1}$, and $R_p = \frac{\mathbb{Z}_p[x]}{x^N - 1}$ which have the form $\mathbf{a}(x) \in \frac{\mathbb{Z}_k[x]}{x^N - 1}$, $\mathbf{a}(x) = a_0 + a_1 x + \cdots + a_{N-1} a^{N-1}$ with coefficients in $\mathbb{Z}_k$.

Convolution Polynomial Rings

That is, NTRU operates on convolution polynomial rings $R = \frac{\mathbb{Z}[x]}{x^N - 1}$, $R_q = \frac{\mathbb{Z}_q[x]}{x^N - 1}$, and $R_p = \frac{\mathbb{Z}_p[x]}{x^N - 1}$ which have the form $\mathbf{a}(x) \in \frac{\mathbb{Z}_k[x]}{x^N - 1}$, $\mathbf{a}(x) = a_0 + a_1 x + \cdots + a_{N-1} a^{N-1}$ with coefficients in $\mathbb{Z}_k$.

Most significantly, $x^N \equiv 1 \bmod (x^N - 1)$.

Convolution Polynomial Rings

That is, NTRU operates on convolution polynomial rings $R = \frac{\mathbb{Z}[x]}{x^N-1}$, $R_q = \frac{\mathbb{Z}_q[x]}{x^N-1}$, and $R_p = \frac{\mathbb{Z}_p[x]}{x^N-1}$ which have the form $\mathbf{a}(x) \in \frac{\mathbb{Z}_k[x]}{x^N-1}$, $\mathbf{a}(x) = a_0 + a_1 x + \cdots + a_{N-1} a^{N-1}$ with coefficients in $\mathbb{Z}_k$.

Most significantly, $x^N \equiv 1 \bmod (x^N - 1)$.

Out of convenience, we can write $\mathbf{a}(x) \sim (a_0, \cdots, a_{N-1}) \in \mathbb{Z}^N$.

Convolution Polynomial Rings

That is, NTRU operates on convolution polynomial rings $R = \frac{\mathbb{Z}[x]}{x^N-1}$, $R_q = \frac{\mathbb{Z}_q[x]}{x^N-1}$, and $R_p = \frac{\mathbb{Z}_p[x]}{x^N-1}$ which have the form $\mathbf{a}(x) \in \frac{\mathbb{Z}_k[x]}{x^N-1}$, $\mathbf{a}(x) = a_0 + a_1 x + \cdots + a_{N-1} a^{N-1}$ with coefficients in $\mathbb{Z}_k$.

Most significantly, $x^N \equiv 1 \bmod (x^N - 1)$.

Out of convenience, we can write $\mathbf{a}(x) \sim (a_0, \cdots, a_{N-1}) \in \mathbb{Z}^N$.

Addition is defined normally with $\mathbf{a}(x) + \mathbf{b}(x) = (a_0 + b_0, \cdots, a_{N-1} + b_{N-1})$.

Convolution Polynomial Rings

That is, NTRU operates on convolution polynomial rings $R = \frac{\mathbb{Z}[x]}{x^N - 1}$, $R_q = \frac{\mathbb{Z}_q[x]}{x^N - 1}$, and $R_p = \frac{\mathbb{Z}_p[x]}{x^N - 1}$ which have the form $\mathbf{a}(x) \in \frac{\mathbb{Z}_k[x]}{x^N - 1}$, $\mathbf{a}(x) = a_0 + a_1 x + \cdots + a_{N-1} a^{N-1}$ with coefficients in $\mathbb{Z}_k$.

Most significantly, $x^N \equiv 1 \bmod (x^N - 1)$.

Out of convenience, we can write $\mathbf{a}(x) \sim (a_0, \cdots, a_{N-1}) \in \mathbb{Z}^N$.

Addition is defined normally with $\mathbf{a}(x) + \mathbf{b}(x) = (a_0 + b_0, \cdots, a_{N-1} + b_{N-1})$.

Multiplication is... more complicated...

Convolution Polynomial Rings (cont.)

Multiplication is the usual but $x^2 * x^2 \mod (x^3 - 1) \equiv x$.

Convolution Polynomial Rings (cont.)

Multiplication is the usual but $x^2 * x^2 \bmod (x^3 - 1) \equiv x$.

We can simplify this by writing this explicitly:

Convolution Polynomial Rings (cont.)

Multiplication is the usual but $x^2 * x^2 \bmod (x^3 - 1) \equiv x$.

We can simplify this by writing this explicitly:

$\mathbf{a}(x) \star \mathbf{b}(x) = \mathbf{c}(x)$ with $c_k = \sum_{i+j \equiv k \bmod N} a_i b_{k-i}$

Convolution Polynomial Rings (cont.)

Multiplication is the usual but $x^2 * x^2 \bmod (x^3 - 1) \equiv x$.

We can simplify this by writing this explicitly:

$\mathbf{a}(x) \star \mathbf{b}(x) = \mathbf{c}(x)$ with $c_k = \sum_{i+j \equiv k \bmod N} a_i b_{k-i}$

Example: $\mathbf{a}(x) = (1, -2, 0, 4, -1)$, $\mathbf{b}(x) = (3, 4, -2, 5, 4)$.
$\mathbf{a}(x) \star \mathbf{b}(x) = (-13, 20, -7, 19, 5)$.

Relationship between R and R_q

We can define a ring homomorphism from $R \to R_q$ by reducing coefficients in R modulo q.

Relationship between R and R_q

We can define a ring homomorphism from $R \to R_q$ by reducing coefficients in R modulo q.

We *cannot* easily do the same for $R_q \to R$. Instead, when $\mathbf{a}(x) \in R_q$, we define the *centered lift of* $\mathbf{a}(x)$ *to* R to be the unique polynomial $\mathbf{a}'(x) \in R$ such that $\mathbf{a}'(x) \bmod q = \mathbf{a}(x)$ such that $-\frac{q}{2} < a'_i \leq \frac{q}{2}$ for all i.

One More Thing

One More Thing

We need one last bit of notation to describe the NTRU system concisely.

One More Thing

We need one last bit of notation to describe the NTRU system concisely.

Given d_1, d_2, positive integers,

$$\tau(d_1, d_2) = \left\{ \mathbf{a}(x) \in R : \begin{array}{r} \mathbf{a}(x) \text{ has } d_1 \text{ coefficients equal to } 1 \\ \mathbf{a}(x) \text{ has } d_2 \text{ coefficients equal to -1} \\ \mathbf{a}(x) \text{ has all other coefficients } 0 \end{array} \right\}$$

NTRU

NTRU

1. Alice chooses (N, p, q, d) with $N,\ p$ prime, $gcd(N, q) = \text{gcd}(p, q) = 1$, and $q > (6d + 1)p$.

NTRU

1. Alice chooses (N, p, q, d) with N, p prime, $gcd(N, q) = \mathrm{gcd}(p, q) = 1$, and $q > (6d + 1)p$.
2. Alice chooses $\mathbf{f} \in \tau(d + 1, d)$ and $\mathbf{g} \in \tau(d, d)$.

NTRU

1. Alice chooses (N, p, q, d) with $N,\ p$ prime, $gcd(N, q) = \gcd(p, q) = 1$, and $q > (6d + 1)p$.
2. Alice chooses $\mathbf{f} \in \tau(d + 1, d)$ and $\mathbf{g} \in \tau(d, d)$.
3. Alice computes $\mathbf{F_q} = f^{-1} \in R_q$ and $\mathbf{F_p} = f^{-1} \in R_p$ and publishes her public key (N, p, q, d, h), with $\mathbf{h} = \mathbf{F_q} \star \mathbf{g}$.

NTRU

1. Alice chooses (N, p, q, d) with N, p prime, $gcd(N, q) = \text{gcd}(p, q) = 1$, and $q > (6d + 1)p$.
2. Alice chooses $\mathbf{f} \in \tau(d + 1, d)$ and $\mathbf{g} \in \tau(d, d)$.
3. Alice computes $\mathbf{F_q} = f^{-1} \in R_q$ and $\mathbf{F_p} = f^{-1} \in R_p$ and publishes her public key (N, p, q, d, h), with $\mathbf{h} = \mathbf{F_q} \star \mathbf{g}$.
4. Bob chooses $\mathbf{m} \in R_p$ and an $\mathbf{r} \in \tau(d, d)$ and sends Alice $\mathbf{c} \equiv p\mathbf{r} \star \mathbf{h} + \mathbf{m} \mod q$.

NTRU

1. Alice chooses (N, p, q, d) with $N,\ p$ prime, $gcd(N, q) = \gcd(p, q) = 1$, and $q > (6d + 1)p$.
2. Alice chooses $\mathbf{f} \in \tau(d + 1, d)$ and $\mathbf{g} \in \tau(d, d)$.
3. Alice computes $\mathbf{F_q} = f^{-1} \in R_q$ and $\mathbf{F_p} = f^{-1} \in R_p$ and publishes her public key (N, p, q, d, h), with $\mathbf{h} = \mathbf{F_q} \star \mathbf{g}$.
4. Bob chooses $\mathbf{m} \in R_p$ and an $\mathbf{r} \in \tau(d, d)$ and sends Alice $\mathbf{c} \equiv p\mathbf{r} \star \mathbf{h} + \mathbf{m} \mod q$.
5. Alice computes $\mathbf{a} = \mathbf{f} \star \mathbf{c}$ and the centered lift of $\mathbf{a}$, which is $\mathbf{a}'$.

NTRU

1. Alice chooses (N, p, q, d) with N, p prime, $gcd(N, q) = \gcd(p, q) = 1$, and $q > (6d + 1)p$.
2. Alice chooses $\mathbf{f} \in \tau(d + 1, d)$ and $\mathbf{g} \in \tau(d, d)$.
3. Alice computes $\mathbf{F_q} = f^{-1} \in R_q$ and $\mathbf{F_p} = f^{-1} \in R_p$ and publishes her public key (N, p, q, d, h), with $\mathbf{h} = \mathbf{F_q} \star \mathbf{g}$.
4. Bob chooses $\mathbf{m} \in R_p$ and an $\mathbf{r} \in \tau(d, d)$ and sends Alice $\mathbf{c} \equiv p\mathbf{r} \star \mathbf{h} + \mathbf{m} \mod q$.
5. Alice computes $\mathbf{a} = \mathbf{f} \star \mathbf{c}$ and the centered lift of $\mathbf{a}$, which is $\mathbf{a}'$.
6. Alice decrypts the message by computing $\mathbf{m} \equiv \mathbf{F_p} \star \mathbf{a}' \mod p$.

NTRU

1. Alice chooses (N, p, q, d) with N, p prime, $gcd(N, q) = \gcd(p, q) = 1$, and $q > (6d + 1)p$.
2. Alice chooses $\mathbf{f} \in \tau(d + 1, d)$ and $\mathbf{g} \in \tau(d, d)$.
3. Alice computes $\mathbf{F_q} = f^{-1} \in R_q$ and $\mathbf{F_p} = f^{-1} \in R_p$ and publishes her public key (N, p, q, d, h), with $\mathbf{h} = \mathbf{F_q} \star \mathbf{g}$.
4. Bob chooses $\mathbf{m} \in R_p$ and an $\mathbf{r} \in \tau(d, d)$ and sends Alice $\mathbf{c} \equiv p\mathbf{r} \star \mathbf{h} + \mathbf{m} \mod q$.
5. Alice computes $\mathbf{a} = \mathbf{f} \star \mathbf{c}$ and the centered lift of $\mathbf{a}$, which is $\mathbf{a}'$.
6. Alice decrypts the message by computing $\mathbf{m} \equiv \mathbf{F_p} \star \mathbf{a}' \mod p$.

Example

NTRU as a Lattice

NTRU as a Lattice

The NTRU Lattice is generated by the rows of the block matrix

$$M_{\mathbf{h}}^{NTRU} = \begin{pmatrix} 1 & \mathbf{h} \\ 0 & q \end{pmatrix} \text{ where } \mathbf{h} = \begin{pmatrix} h_0 & h_1 & \cdots & h_{N-1} \\ h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & \cdots & h_0 \end{pmatrix}.$$

NTRU as a Lattice

The NTRU Lattice is generated by the rows of the block matrix

$$M_{\mathbf{h}}^{NTRU} = \begin{pmatrix} 1 & \mathbf{h} \\ 0 & q \end{pmatrix} \text{ where } \mathbf{h} = \begin{pmatrix} h_0 & h_1 & \cdots & h_{N-1} \\ h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & \cdots & h_0 \end{pmatrix}.$$

We can see this because $\mathbf{h} \equiv \mathbf{f}^{-1}\mathbf{g} \bmod q \implies \mathbf{f} \star \mathbf{h} = \mathbf{g} + q\mathbf{u}$.

NTRU as a Lattice

The NTRU Lattice is generated by the rows of the block matrix

$$M_{\mathbf{h}}^{NTRU} = \begin{pmatrix} 1 & \mathbf{h} \\ 0 & q \end{pmatrix} \text{ where } \mathbf{h} = \begin{pmatrix} h_0 & h_1 & \cdots & h_{N-1} \\ h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & \cdots & h_0 \end{pmatrix}.$$

We can see this because $\mathbf{h} \equiv \mathbf{f}^{-1}\mathbf{g} \bmod q \implies \mathbf{f} \star \mathbf{h} = \mathbf{g} + q\mathbf{u}$. Then $(\mathbf{f}, \mathbf{g}) \in L_{\mathbf{h}}^{NTRU}$ because

$$(\mathbf{f}, -\mathbf{u}) \begin{pmatrix} 1 & \mathbf{h} \\ 0 & q \end{pmatrix} = (\mathbf{f}, \mathbf{f} \star \mathbf{h} - q\mathbf{u}) = (\mathbf{f}, \mathbf{g})$$

NTRU as a Lattice

The NTRU Lattice is generated by the rows of the block matrix

$$M_{\mathbf{h}}^{NTRU} = \begin{pmatrix} 1 & \mathbf{h} \\ 0 & q \end{pmatrix} \text{ where } \mathbf{h} = \begin{pmatrix} h_0 & h_1 & \cdots & h_{N-1} \\ h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & \cdots & h_0 \end{pmatrix}.$$

We can see this because $\mathbf{h} \equiv \mathbf{f}^{-1}\mathbf{g} \bmod q \implies \mathbf{f} \star \mathbf{h} = \mathbf{g} + q\mathbf{u}$. Then $(\mathbf{f}, \mathbf{g}) \in L_{\mathbf{h}}^{NTRU}$ because

$$(\mathbf{f}, -\mathbf{u}) \begin{pmatrix} 1 & \mathbf{h} \\ 0 & q \end{pmatrix} = (\mathbf{f}, \mathbf{f} \star \mathbf{h} - q\mathbf{u}) = (\mathbf{f}, \mathbf{g})$$

Moreover, $||(\mathbf{f}, \mathbf{g})|| \approx \sqrt{4d} \approx 1.155\sqrt{N} < \sigma(L_{\mathbf{h}}^{NTRU}) \approx 0.484N$.

Lattice Attacks on NTRU

The best known attack on **NTRUEncrypt** is a hybrid attack.

Lattice Attacks on NTRU

The best known attack on **NTRUEncrypt** is a hybrid attack.

1. Reduce the basis of the lattice for some time as quickly as possible (LLL). (With sufficiently small N or enough time, this is enough to solve the system alone.)

Lattice Attacks on NTRU

The best known attack on **NTRUEncrypt** is a hybrid attack.

1. Reduce the basis of the lattice for some time as quickly as possible (LLL). (With sufficiently small N or enough time, this is enough to solve the system alone.)
2. We perform a meet-in-the-middle, or collision algorithm, attack on the reduced lattice.

Lattice Attacks on NTRU

The best known attack on **NTRUEncrypt** is a hybrid attack.

1. Reduce the basis of the lattice for some time as quickly as possible (LLL). (With sufficiently small N or enough time, this is enough to solve the system alone.)
2. We perform a meet-in-the-middle, or collision algorithm, attack on the reduced lattice.
3. Profit.

Example Results:
Using a small example with $N = 53, q = 36, d_f = d_g = 16$, a standard meet-in-the-middle attack should take $2^{20.1}$ steps whereas the hybrid attack used $2^{13.1}$ steps.

That's it for now!

Explaining LLL and demonstrating an attack on NTRU using LLL is another project entirely. :)

References

Jeffrey Hoffstein - Jill Catherine Pipher - Joseph H.Silverman - Springer - 2008

N. Howgrave-Graham, *A hybrid meet-in-the-middle and lattice reduction attack on NTRU*, pp.150-169, CRYPTO 2007