NTRU - Test for new deployment code to be usable by you and others in the future

Have you ever spent days **trying to repeat the results from few weeks or months ago**? Or you have to do paper revisions, but you just can't get the results to match up? It's unpleasant for both you and science.

In this lesson we will explore different methods and tools for better reproducibility in research software and data. We will demonstrate how version control, workflows, containers, and package managers can be used to **record reproducible environments and computational steps** for our future selves and others.

Learning outcomes

By the end of this lesson, learners should:

- Be able to apply well organized directory structure for their project
- Understand that code can have dependencies, and know how to document them
- Be able to document computational steps, and have an idea when it can be useful
- Know about use cases for containers

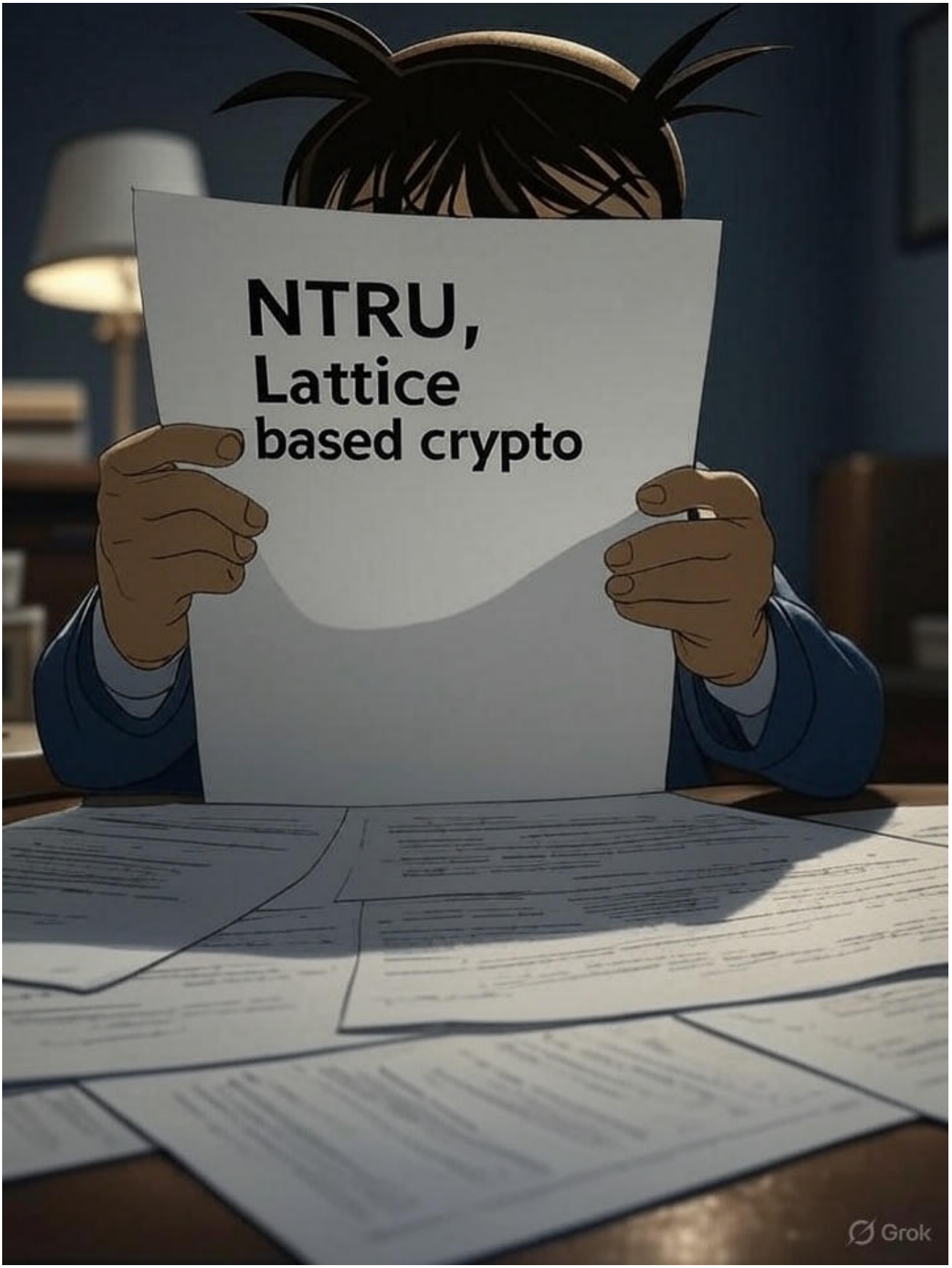
⚙ Prerequisites

You need to install Git, Python, and Snakemake.

If you wish to follow in the terminal and are new to the command line, we recorded a short shell crash course.

How to use

Will edit later



Lattice

Definition (Lattice).
Let B be a $d\times m$ matrix with d linearly independent rows $\{\mathbf{b}_0,\mathbf{b}_1,\ldots,\mathbf{b}_{d-1}\}\subset \mathbb{R}^m$.
The lattice generated by B is defined as

$$\mathcal{L}_B = \mathbb{Z}^d B = \left\{ \sum_{i=0}^{d-1} \gamma_i \mathbf{b}_i : \gamma_i \in \mathbb{Z} \right\}.$$

The matrix B is called the **basis matrix** for the lattice $\mathcal{L}_B$.
Here, d, i.e., the number of linearly independent rows in the basis matrix, is called the *rank*,
and m is called the *dimension* of $\mathcal{L}_B$.

The lattice is referred to as *full-rank* if $d = m$.
If $\mathbf{b}_i \in \mathbb{Z}^m$, we call the lattice an *integral lattice*.
For this work, we consider full-rank integral lattices.

The **volume** of a lattice $\mathcal{L}_B$ defined by a basis matrix B is given by

$$\mathrm{vol}(\mathcal{L}_B) = \sqrt{\lvert \det(BB^{\mathsf{T}}) \rvert},$$

and it is independent of the choice of basis.

For $i \in \{0,1,\ldots,d-1\}$, define π_i to be the projection onto the space orthogonal to the span of $\{\mathbf{b}_0, \ldots, \mathbf{b}_{i-1}\}$,
and denote the **Gram–Schmidt basis** as

$$\{\mathbf{b}_0^\star, \mathbf{b}_1^\star, \ldots, \mathbf{b}_{d-1}^\star\}, \quad \mathbf{b}_i^\star = \pi_i(\mathbf{b}_i).$$

We refer to the lattice generated from $\{ \pi_\ell(\mathbf{b}_\ell), \ldots, \pi_\ell(\mathbf{b}_{r-1})\}$ as the **projected sublattice**, denoted by $\mathcal{L}_{B[\ell,r)}$.

We refer to the lengths $\lVert \mathbf{b}_i^\star \rVert$ for $i \in \{0,1,\ldots,d-1\}$ as the **profile** of the basis B.

Definition (q-ary lattice).
A lattice of dimension d is called a *q-ary lattice* if

$$q\mathbb{Z}^d \subseteq \mathcal{L}_B$$

for some $q > 0$.

Definition (Minimum length).
The minimum length $\lambda_1(\mathcal{L})$ of a lattice $\mathcal{L}$ is defined as the length of its shortest nonzero vector:

$$\lambda_1(\mathcal{L}) := \min_{\mathbf{v} \in \mathcal{L} \setminus \{0\}} \lVert \mathbf{v} \rVert.$$

Definition (Gaussian heuristic).
Given a random d-dimensional lattice $\mathcal{L}_B$ defined by basis B, the Gaussian heuristic estimates that the expected length of the shortest nonzero vector in

(\mathcal{L}_B\) is

$$\lambda_1(\mathcal{L}_B) \;\approx\; \sqrt{\tfrac{d}{2 \pi e}} \, \mathrm{vol}(\mathcal{L}_B)^{\tfrac{1}{d}}.$$

Definition (Hard lattice problems)

Let \(\mathcal{L}_B \subset \mathbb{R}^d\) be a full-rank lattice defined by the basis \(B\).

1. **Shortest Vector Problem (SVP):**
   Find a nonzero vector \(\mathbf{v} \in \mathcal{L}_B\) such that
   \(\|\mathbf{v}\| = \lambda_1(\mathcal{L}_B)\).
2. **Closest Vector Problem (CVP):**
   Find a vector \(\mathbf{v} \in \mathcal{L}_B\) closest to the given target vector \(\mathbf{t} \in \mathbb{R}^d\), i.e.,
   \(\|\mathbf{v} - \mathbf{t}\| \leq \|\mathbf{w} - \mathbf{t}\|\) for all \(\mathbf{w} \in \mathcal{L}_B\).

   Further, when \(\|\mathbf{v} - \mathbf{t}\| < \alpha \lambda_1(\mathcal{L}_B)\) for some \(\alpha < 1\),
   the problem is referred to as the **Bounded Distance Decoding (BDD)** problem.

NTRU

Definition (NTRU)

Since its introduction in 1998, several versions of NTRU have emerged in the literature. NTRU is now recognized as a hard problem in cryptography rather than a unique cryptosystem that can be extended to different algebraic structures.
The NTRU design and the problem can be outlined as:

NTRU:
Let \(N\) be a prime, \(q\) be a positive integer, and \(f,g \in \mathbb{Z}[x]/\langle x^N-1\rangle\) be two polynomials with small coefficients (mostly ternary) such that \(f\) is invertible modulo \(q\).
The pair \((f,g)\) forms the secret key and

$$h = f^{-1} \ast g \pmod{q} \in \mathbb{Z}_q[x]/\langle x^N-1\rangle$$

is the public key. The NTRU problem asks to find the private key or its rotations \((x^i \ast f, x^i \ast g)\).

The most renowned technique to attack the NTRU problem is to solve SVP in the \(2N\)-dimensional lattice \(\Lambda_{CS}\) generated by the basis

$$\begin{split} \mathcal{M}_{CS} = \begin{pmatrix} I_N & \mathcal{H} \\ 0_N & qI_N \end{pmatrix}, \end{split}$$

since the vector $(\mathbf{f}, \mathbf{g})$ associated with the private key (f,g) or its rotations are the shortest vectors in the lattice Λ_{CS} with high probability.

NTRU Lattice

Out of all the attacks proposed on NTRU, the lattice based attack is by far the best in present day literature.

NTRU Public Key

Let $T(r_1,r_2) = \lbrace f \in \frac{\mathbb{Z}[X]}{<X^N-1>} \rbrace$ such that r_1 coefficients in f are 1, r_2 coefficients in f are -1 and remaining coefficients in f are 0. The set $T(r_1,r_2)$ is known as the set of ternary polynomials, as the coefficients of these polynomias only take 3 different values.
Now let $d = \lfloor \frac{N}{3} \rfloor$. It is observed that with high probability, the elements of $T(d,d)$ are invertible in $\frac{\mathbb{Z}_q[X]}{<X^N-1>}$ and $\frac{\mathbb{Z}_p[X]}{<X^N-1>}$.
Let $f \in T(d,d)$ and $g \in T(d+1,d)$.
Let $h = f^{-1} * g$ (mod q).
The polynomial h is the public key.

Lattice construction from Public Key

Let $h = f^{-1} * g$ (mod q) be a Public Key for NTRU cryptosystem.
Then, $g = f*h$ (mod q) $\Rightarrow g = f*h + q*u$, for some $u \in \frac{\mathbb{Z}[X]}{<X^N-1>}$.
Also, $f = 1*f + 0*u$.
Thus, consider the matrix equation,
$\begin{bmatrix} f_0 & f_1 & \cdots & f_{N-1} & g_0 & g_1 & \cdots & g_{N-1} \end{bmatrix} =$
$\begin{bmatrix} f_0 & f_1 & \cdots & f_{N-1} & u_0 & u_1 & \cdots & u_{N-1} \end{bmatrix} * \begin{bmatrix} 1 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & 1 & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{N-2} \\ \cdots \\ 0 & 0 & \cdots & 1 & h_1 & h_2 & \cdots & h_0 \\ 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \cdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \\ \end{bmatrix}$.
This equation represents the matrix form of the Public Key equations. In a more compact form one can write,
$\begin{bmatrix} f & g \end{bmatrix} = \begin{bmatrix} f & u \end{bmatrix} * L$,
where $L = \begin{bmatrix} I_N & H \\ 0_N & qI_N \end{bmatrix}$ is a matrix of dimension $2N \times 2N$,
I_N is the identity matrix of dimension $N \times N$ and
0_N is the null matrix of dimension $N \times N$ and
$H = \begin{bmatrix} h_0 & h_1 & \cdots & h_{N-1} \\ h_{N-1} & h_0 & \cdots & h_{N-2} \\ \cdots \\ h_1 & h_2 & \cdots & h_0 \\ \end{bmatrix}$

Now, note that $\begin{bmatrix} f & g \end{bmatrix}$ is a short vector in L.
Thus, finding f and g from h is equivalent to finding short vector in L.

GR-NTRU (Group Ring NTRU)

It is observed that the Group-ring structure is very useful for generalizing and creating lattices for NTRU and NTRU-like cryptosystems.

Group Ring Definition

Let R be a ring and $G = \lbrace g_1,g_2,\ldots,g_n \rbrace$ be a finite group of order n. Then, the group ring of G over R is.
$RG = \lbrace \sum\limits_{i=1}^{n}r_{g_i}g_i \mid r_{g_i} \in R, 1 \leq i \leq n \rbrace$
The addition and multiplication operations in the GR are given as follows-
$r+s = \sum\limits_{i=1}^{n} (r_{g_i} + s_{g_i})g_i$
$r*s = \sum\limits_{i=1}^{n} (\sum\limits_{g_h * g_k = g_i} r_{g_h}s_{g_k})g_i$

NTRU as a Group-ring

Another perspective of NTRU defined over the ring $\mathbb{Z}[X]/<X^N-1>$ is to consider it as the Group-ring $\mathbb{Z}C_N$, where $C_N = \lbrace 1,x,x^2,...x^{N-1} \rbrace$ is the cyclic group of order N.
Let $R = \mathbb{Z}$, ring of integers. Then the following changes convert NTRU to a Group-ring.
$\frac{\mathbb{Z}[X]}{<X^N-1>} \cong \mathbb{Z}C_N$
$\frac{\mathbb{Z}_q[X]}{<X^N-1>} \cong \mathbb{Z}_qC_N$
$\frac{\mathbb{Z}_p[X]}{<X^N-1>} \cong \mathbb{Z}_pC_N$

Twisted GR-NTRU

Will edit later

Group-ring Lattice

The Group-ring interpretetion can also be used to construct Lattice for NTRU-like cryptosystems.

Lattice from Group-ring

Let $G = \lbrace a_1,a_2,\cdots a_n \rbrace$ be a group and R be a ring. R is preferably commutative with identity.
The choice of such R is so that few invertible elements may be present in R.
Let RG be the group ring. Let $f,g \in RG$ such that f is invertible.
Let $h = f^{-1} * g$ be the Public Key.
Then, $g = f * h$. We can construct similar matrix equations as we did for NTRU.

$f = f * 1 + u * 0$ and $g = f * h + u * q$.
$H = \begin{bmatrix} h_{a_1^{-1}a_1} & h_{a_1^{-1}a_2} & \cdots & h_{a_1^{-1}a_n} \\ h_{a_2^{-1}a_1} & h_{a_2^{-1}a_2} & \cdots & h_{a_2^{-1}a_n} \\ \cdots \\ h_{a_n^{-1}a_1} & h_{a_2^{-1}a_2} & \cdots & h_{a_n^{-1}a_n} \end{bmatrix}$
$\begin{bmatrix} f & g \end{bmatrix} = \begin{bmatrix} f & u \end{bmatrix} * \begin{bmatrix} I_n & H \\ 0_n & qI_n \end{bmatrix}$.

However if the multiplication is from right, i.e., f and g are column vectors instead of row vectors,
then the equation and structure of H changs a little.
$g = H * f + q * u$ and $H = [h_{a_ia_j^{-1}}]$ instead of $H = [h_{a_i^{-1}a_j}]$.

Lattice from twisted Group-ring

Will edit later.
Will edit later

Primal Attack

Will edit later

Hybrid Attack

Will edit later

Sub-field Attack

Will edit later

Example

Will edit later