All Along the Ring Tower

Algebraic Structures for Fun and Profit

Thomas Prest
joint work w/
{Léo Ducas} $\cup$ {Thomas Pornin} $\cup$ {Léo Ducas, Steven Galbraith, Yang Yu}

RISC $\times$ PROMETHEUS Seminar, 03/05/2019

Rings in Lattice-Based Cryptography

It is typical in lattice-based cryptography to use matrices with coefficients in $\mathbb{Z}_q[x]/(x^d+1)$ rather than $\mathbb{Z}_q$:

1. Communication costs typically go $O(d^2) \Rightarrow O(d)$
2. Computation costs typically go $O(d^2) \Rightarrow O(d \log d)$

But in some situations this additional structure seems ineffective:

1. Matrix decomposition (Cholesky, Gram-Schmidt, etc.)
2. Solving equations in a ring which is not a field (e.g. $\mathbb{Z}[x]/(x^d+1)$)

Algorithms can take time up to $\Theta(d^2)$ or $\Theta(d^3)$.

The State of Affairs

What naïve solutions do:

1. View $\mathbb{Q}[x]/(x^d + 1)$ as either a $\mathbb{Q}$-linear space of dimension d, an extension field of $\mathbb{Q}$ of degree d, etc.
2. This ignores the rich structure of cyclotomic rings and fields.

What happens when we open the black box?

Cyclotomic Fields - Opening the Black Box

For d a power-of-two, we note:

- $\mathcal{Q}_d = \mathbb{Q}[x]/(x^d + 1)$ the d-th *cyclotomic field*
- $\mathcal{Z}_d = \mathbb{Z}[x]/(x^d + 1)$ the d-th *cyclotomic ring*

We have this tower of fields:

$$\mathbb{Q} \subsetneq \mathcal{Q}_2 \subsetneq \cdots \subseteq \mathcal{Q}_{d/2} \subsetneq \mathcal{Q}_d$$

As well as this chain of isomorphisms:

$$\mathbb{Q}^d \cong (\mathcal{Q}_2)^{d/2} \cong \ldots \cong (\mathcal{Q}_{d/2})^2 \cong \mathcal{Q}_d$$

At a high level:

- The **field norm** and **field trace** allows to move in the tower of fields
- **Ring isomorphisms** allow us to move in the chain of ring isomorphisms

Traces and Norms in Cyclotomic Fields

Definition: For a (finite) field extension L/K:

- The field trace is:

$$\mathrm{Tr}_{L/K} : \quad L \to K, \qquad f \mapsto \sum_{\sigma \in \mathrm{Gal}(L/K)} \sigma(f)$$

- The field norm is:

$$\mathrm{N}_{L/K} : \quad L \to K, \qquad f \mapsto \prod_{\sigma \in \mathrm{Gal}(L/K)} \sigma(f)$$

Concretely: if $f(x) = f_e(x^2) + x \cdot f_o(x^2) \in \mathcal{Q}_d$, then $f^\times(x) = f(-x)$ and:

- $\mathrm{Tr}_{\mathcal{Q}_d/\mathcal{Q}_{d/2}}(f) = f + f^\times = 2 \cdot f_e(x^2)$
- $\mathrm{N}_{\mathcal{Q}_d/\mathcal{Q}_{d/2}}(f) = f \cdot f^\times = f_e^2(x^2) - x^2 f_o^2(x^2)$

Composition properties:

- $\mathrm{Tr}_{L/K} \circ \mathrm{Tr}_{M/L} = \mathrm{Tr}_{M/K}$
- $\mathrm{N}_{L/K} \circ \mathrm{N}_{M/L} = \mathrm{N}_{M/K}$

Homomorphic properties:

- $\mathrm{Tr}_{L/K}(a+b) = \mathrm{Tr}_{L/K}(a) + \mathrm{Tr}_{L/K}(b)$
- $\mathrm{N}_{L/K}(a \cdot b) = \mathrm{N}_{L/K}(a) \cdot \mathrm{N}_{L/K}(b)$

Problem 1 - Completing NTRU Bases

NTRU Lattices:

- Prevalent in lattice-based crypto
- Public key is $\mathbf{A} = \left[\begin{array}{c|c} 1 & h \end{array}\right]$, for $h = g \times f^{-1} \bmod (\varphi, q)$.
- Private key is $\mathbf{B}$ such that $\mathbf{B} \times \mathbf{A}^t = 0 \bmod (\varphi, q)$

Some schemes only require a partial trapdoor $\mathbf{B} = \left[\begin{array}{c|c} g & -f \end{array}\right]$:

- Fiat-Shamir [ZCHW17], encryption [SHRS17], FHE [LTV12, BLLN13]

Problem 1 - Completing NTRU Bases

NTRU Lattices:

- Prevalent in lattice-based crypto
- Public key is $\mathbf{A} = \left[\begin{array}{c|c} 1 & h \end{array}\right]$, for $h = g \times f^{-1} \bmod (\varphi, q)$.
- Private key is $\mathbf{B}$ such that $\mathbf{B} \times \mathbf{A}^t = 0 \bmod (\varphi, q)$

Some schemes only require a partial trapdoor $\mathbf{B} = \left[\begin{array}{c|c} g & -f \end{array}\right]$:

- Fiat-Shamir [ZCHW17], encryption [SHRS17], FHE [LTV12, BLLN13]

However, some schemes require a full trapdoor $\mathbf{B} = \left[\begin{array}{c|c} g & -f \\ \hline G & -F \end{array}\right]$:

- Hash-then-sign [PFH$^+$17], IBE [DLP14], HIBE [CG17]
- More generally, anything based on trapdoor sampling [GPV08]

Problem 1 - Completing NTRU Bases

NTRU Lattices:

- Prevalent in lattice-based crypto
- Public key is $\mathbf{A} = \left[\begin{array}{c|c} 1 & h \end{array}\right]$, for $h = g \times f^{-1} \bmod (\varphi, q)$.
- Private key is $\mathbf{B}$ such that $\mathbf{B} \times \mathbf{A}^t = 0 \bmod (\varphi, q)$

Some schemes only require a partial trapdoor $\mathbf{B} = \left[\begin{array}{c|c} g & -f \end{array}\right]$:

- Fiat-Shamir [ZCHW17], encryption [SHRS17], FHE [LTV12, BLLN13]

However, some schemes require a full trapdoor $\mathbf{B} = \left[\begin{array}{c|c} g & -f \\ \hline G & -F \end{array}\right]$:

- Hash-then-sign [PFH+17], IBE [DLP14], HIBE [CG17]
- More generally, anything based on trapdoor sampling [GPV08]

Problem: Given $f, g \in \mathbb{Z}[x]/(x^d+1)$, find $F, G \in \mathbb{Z}[x]/(x^d+1)$ such that:

$$f \cdot G - g \cdot F = q$$

Fun fact

If we can solve the problem projected over $\mathcal{Z}_{d/2}$, i.e.:

$$\mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(f) \cdot G' - \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(g) \cdot F' = 1$$

for some F', G', then we have this relationship over $\mathcal{Z}_d$:

$$f \cdot (f^{\times} G') - g \cdot (g^{\times} F') = 1$$

This leads to a simple algorithm:

1. Project
2. Solve
3. Lift

Outline of the Solver

$$\begin{array}{ccc}
\mathcal{Z}_d & \ni & f, g \\
\cup\nmid & & \\
\mathcal{Z}_{d/2} & & \\
\cup\nmid & & \\
\mathcal{Z}_{d/4} & & \\
\cup\nmid & & \\
\vdots & & \\
\cup\nmid & & \\
\mathbb{Z} & &
\end{array}$$

Outline of the Solver

$$
\begin{array}{ccc}
\mathcal{Z}_d & \ni & f, g \\
\cup\!\! \shortmid & & \downarrow \\
\mathcal{Z}_{d/2} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(g) \\
\cup\!\! \shortmid & & \\
\mathcal{Z}_{d/4} & & \\
\cup\!\! \shortmid & & \\
\vdots & & \\
\cup\!\! \shortmid & & \\
\mathbb{Z} & &
\end{array}
$$

Outline of the Solver

$$
\begin{array}{ccc}
\mathcal{Z}_d & \ni & f, g \\
\rotatebox{90}{$\subsetneq$} & & \downarrow \\
\mathcal{Z}_{d/2} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(g) \\
\rotatebox{90}{$\subsetneq$} & & \downarrow \\
\mathcal{Z}_{d/4} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(g) \\
\rotatebox{90}{$\subsetneq$} & & \\
\vdots & & \\
\rotatebox{90}{$\subsetneq$} & & \\
\mathbb{Z} & &
\end{array}
$$

Outline of the Solver

$$
\begin{array}{ccc}
\mathcal{Z}_d & \ni & f, g \\
\cup\!\!\nmid & & \downarrow \\
\mathcal{Z}_{d/2} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(g) \\
\cup\!\!\nmid & & \downarrow \\
\mathcal{Z}_{d/4} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(g) \\
\cup\!\!\nmid & & \downarrow \\
\vdots & \vdots & \vdots \\
\cup\!\!\nmid & & \\
\mathbb{Z} & &
\end{array}
$$

$$
\begin{array}{ccc}
\mathcal{Z}_d & \ni & f, g \\
\rotatebox{90}{$\supsetneq$} & & \downarrow \\
\mathcal{Z}_{d/2} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(g) \\
\rotatebox{90}{$\supsetneq$} & & \downarrow \\
\mathcal{Z}_{d/4} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(g) \\
\rotatebox{90}{$\supsetneq$} & & \downarrow \\
\vdots & \vdots & \vdots \\
\rotatebox{90}{$\supsetneq$} & & \downarrow \\
\mathbb{Z} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathbb{Z}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathbb{Z}}(g)
\end{array}
$$

Outline of the Solver

$$
\begin{array}{ccccc}
\mathcal{Z}_d & \ni & f, g & & \\
\cup\!\nmid & & \downarrow & & \\
\mathcal{Z}_{d/2} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(g) & & \\
\cup\!\nmid & & \downarrow & & \\
\mathcal{Z}_{d/4} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(g) & & \\
\cup\!\nmid & & \downarrow & & \\
\vdots & \vdots & \vdots & & \\
\cup\!\nmid & & \downarrow & & \\
\mathbb{Z} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathbb{Z}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathbb{Z}}(g) & \to & F^{[\ell]}, G^{[\ell]}
\end{array}
$$

Outline of the Solver

$$
\begin{array}{cccccc}
\mathcal{Z}_d & \ni & f, g & & \\
\rotatebox[origin=c]{90}{$\subsetneq$} & & \downarrow & & \\
\mathcal{Z}_{d/2} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(g) & & \\
\rotatebox[origin=c]{90}{$\subsetneq$} & & \downarrow & & \\
\mathcal{Z}_{d/4} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(g) & & \\
\rotatebox[origin=c]{90}{$\subsetneq$} & & \downarrow & & \\
\vdots & \vdots & \vdots & & \vdots \\
\rotatebox[origin=c]{90}{$\subsetneq$} & & \downarrow & & \uparrow \\
\mathbb{Z} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathbb{Z}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathbb{Z}}(g) & \rightarrow & F^{[\ell]}, G^{[\ell]}
\end{array}
$$

Outline of the Solver

$$
\begin{array}{cccccc}
\mathcal{Z}_d & \ni & f, g & & \\
\rotatebox{90}{$\subsetneq$} & & \downarrow & & \\
\mathcal{Z}_{d/2} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(g) & & \\
\rotatebox{90}{$\subsetneq$} & & \downarrow & & \\
\mathcal{Z}_{d/4} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(g) & \rightarrow & F^{[2]}, G^{[2]} \\
\rotatebox{90}{$\subsetneq$} & & \downarrow & & \uparrow \\
\vdots & \vdots & \vdots & & \vdots \\
\rotatebox{90}{$\subsetneq$} & & \downarrow & & \uparrow \\
\mathbb{Z} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathbb{Z}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathbb{Z}}(g) & \rightarrow & F^{[\ell]}, G^{[\ell]}
\end{array}
$$

Outline of the Solver

$$
\begin{array}{ccccc}
\mathcal{Z}_d & \ni & f, g & & \\
\rotatebox{90}{$\subsetneq$} & & \downarrow & & \\
\mathcal{Z}_{d/2} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(g) & \rightarrow & F^{[1]}, G^{[1]} \\
\rotatebox{90}{$\subsetneq$} & & \downarrow & & \uparrow \\
\mathcal{Z}_{d/4} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(g) & \rightarrow & F^{[2]}, G^{[2]} \\
\rotatebox{90}{$\subsetneq$} & & \downarrow & & \uparrow \\
\vdots & \vdots & \vdots & & \vdots \\
\rotatebox{90}{$\subsetneq$} & & \downarrow & & \uparrow \\
\mathbb{Z} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathbb{Z}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathbb{Z}}(g) & \rightarrow & F^{[\ell]}, G^{[\ell]}
\end{array}
$$

Outline of the Solver

$$
\begin{array}{ccccc}
\mathcal{Z}_d & \ni & f, g & \rightarrow & F, G \\
\cup\!\!\nmid & & \downarrow & & \uparrow \\
\mathcal{Z}_{d/2} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(g) & \rightarrow & F^{[1]}, G^{[1]} \\
\cup\!\!\nmid & & \downarrow & & \uparrow \\
\mathcal{Z}_{d/4} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(g) & \rightarrow & F^{[2]}, G^{[2]} \\
\cup\!\!\nmid & & \downarrow & & \uparrow \\
\vdots & \vdots & \vdots & & \vdots \\
\cup\!\!\nmid & & \downarrow & & \uparrow \\
\mathbb{Z} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathbb{Z}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathbb{Z}}(g) & \rightarrow & F^{[\ell]}, G^{[\ell]}
\end{array}
$$

Outline of the Solver

$$
\begin{array}{cccccc}
\mathcal{Z}_d & \ni & f, g & \rightarrow & F, G \\
\cup\!\nmid & & \downarrow & & \uparrow \\
\mathcal{Z}_{d/2} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/2}}(g) & \rightarrow & F^{[1]}, G^{[1]} \\
\cup\!\nmid & & \downarrow & & \uparrow \\
\mathcal{Z}_{d/4} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathcal{Z}_{d/4}}(g) & \rightarrow & F^{[2]}, G^{[2]} \\
\cup\!\nmid & & \downarrow & & \uparrow \\
\vdots & \vdots & \vdots & & \vdots \\
\cup\!\nmid & & \downarrow & & \uparrow \\
\mathbb{Z} & \ni & \mathrm{N}_{\mathcal{Z}_d/\mathbb{Z}}(f), \mathrm{N}_{\mathcal{Z}_d/\mathbb{Z}}(g) & \rightarrow & F^{[\ell]}, G^{[\ell]}
\end{array}
$$

At each lower level:

- The coefficients grow (in bitsize) by a factor 2...
- ... but the number of coefficients is divided by 2.

Space-saving trick: recompute lazily $\mathrm{N}^i(f), \mathrm{N}^i(g)$ at each step

- Allows a linear time-memory trade-off by a factor $\ell = \log n$

Toy Example

```
sage: f8, g8
-x^7 + 3*x^6 - x^4 + 4*x^3 + 6*x^2 - 2*x - 4,
x^7 - x^6 - 2*x^5 - 4*x^3 - 3*x^2 - x + 7
sage: f4, g4
-51*x^3 + 51*x^2 - 54*x - 17, -33*x^3 - 4*x^2 - 47*x + 57
sage: f2, g2
-2049*x + 3196, -1576*x + 6335
sage: f1, g1
14412817, 42616001
sage: F1, G1
5126443, 15157932
sage: F2, G2
2495*x - 399, 3844*x - 2025
sage: F4, G4
-22*x^3 + 39*x^2 - 23*x - 14, -x^3 - 45*x + 5
sage: F8, G8
-x^7 - x^5 + 3*x^4 + 3*x^3 - 3*x^2 + 4,
2*x^7 - x^6 - x^5 - x^4 - 3*x^3 + x^2 + x - 4
```

Performances

Method	Time complexity[1]	Space complexity[1]
Resultant [HHGP+03]	$\tilde{O}(d(d^2 + B))$	$O(d^2 B)$
HNF [SS11]	$\tilde{O}(d^3 B)$	$O(d^2 B)$
This work (Fast)	$O((dB)^{\log_2 3} \log d)$ [Kara] $\tilde{O}(dB)$ [SchöStr]	$O(d(B + \log d) \log d)$
This work (Compact)	$O((dB)^{\log_2 3} \log^2 d)$ [Kara] $\tilde{O}(dB)$ [SchöStr]	$O(d(B + \log d))$

We gain in practice:

- a factor 100 in memory (3 MB $\rightarrow$ 30 kB)
- a factor 100 in time (2 sec. $\rightarrow$ 20 msec.)

[1] $B = \log_2 ||(f, g)||$

Problem: Given $\mathbf{A} \in \mathcal{R}^{n \times n}$, compute $\mathbf{B}_1, \ldots, \mathbf{B}_k \in \mathcal{R}^{n \times n}$ such that

$$\mathbf{A}\mathbf{A}^{\star} + \sum_i \mathbf{B}\mathbf{B}^{\star} = C \cdot \mathbf{I}_n$$

Algorithmic solutions:

- $\mathcal{R} = \mathbb{R}$, $k = 1$: Cholesky [Pei10]
- $\mathcal{R} = \mathbb{R}[x]/(\varphi)$, $k = 1$: Babylonian method [DN12]
- $\mathcal{R} = \mathbb{Z}$, $k = O(1)$: `ia.cr/2019/320`
- $\mathcal{R} = \mathbb{Z}[x]/(x^d + 1)$, $k = O(\log d)$: This talk + `ia.cr/2019/320`

Simplified Problem

Simplified problem: Given $a \in \mathbb{Z}[x]/(x^d+1)$, compute polynomials $b_1, \ldots, b_{\log_2(d)} \in \mathbb{Z}[x]/(x^d+1)$ such that for some constant C:

$$a\overline{a} + \sum_i b_i \overline{b_i} = C,$$

where $\overline{\cdot}$ denotes the Hermitian adjoint (in our case, $\overline{a}(x) = a(x^{-1})$).

Attempt 1: Galois conjugation and Hermitian adjoint compose nicely:

$$\mathrm{Tr}_{\mathcal{Q}_d/\mathcal{Q}_{d/2}}(a\overline{a}) = a\overline{a} + (a\overline{a})^\times = a\overline{a} + a^\times \overline{a^\times} \in \mathcal{Z}_{d/2}$$

- ☺ We have projected the problem over $\mathcal{Z}_{d/2}$.
- ☹ Unfortunately repeating this trick doesn't scale well.

A Scalable Solution

Attempt 2: Let $a\overline{a} = g$; g is self-adjoint so we can write $g = g_{\text{low}} + \overline{g_{\text{low}}}$. Let $b(x) = 1 - x \cdot g_{o,\text{low}}(x^2)$, then:

$$\begin{aligned} g + b\overline{b} &= g_e(x^2) + x \cdot g_{o,\text{low}}(x^2) + \overline{x \cdot g_{o,\text{low}}(x^2)} \\ &\quad + (1 - x \cdot g_{o,\text{low}}(x^2)) \cdot \overline{(1 - x \cdot g_{o,\text{low}}(x^2))} \\ &= (1 + g_e + g_{o,\text{low}} \cdot \overline{g_{o,\text{low}}})(x^2) \end{aligned}$$

A Scalable Solution

Attempt 2: Let $a\bar{a} = g$; g is self-adjoint so we can write $g = g_{\text{low}} + \overline{g_{\text{low}}}$. Let $b(x) = 1 - x \cdot g_{o,\text{low}}(x^2)$, then:

$$\begin{aligned} g + b\bar{b} &= g_e(x^2) + x \cdot g_{o,\text{low}}(x^2) + \overline{x \cdot g_{o,\text{low}}(x^2)} \\ &\quad + (1 - x \cdot g_{o,\text{low}}(x^2)) \cdot \overline{(1 - x \cdot g_{o,\text{low}}(x^2))} \\ &= (1 + g_e + g_{o,\text{low}} \cdot \overline{g_{o,\text{low}}})(x^2) \end{aligned}$$

- ☺ We have projected the problem over $\mathcal{Z}_{d/2}$.
- ☺ This trick scales well with repetition.
- ☹ It incurs a growth on the coefficients' sizes...

A Scalable Solution

Attempt 2: Let $a\overline{a} = g$; g is self-adjoint so we can write $g = g_{\text{low}} + \overline{g_{\text{low}}}$. Let $b(x) = 1 - x \cdot g_{o,\text{low}}(x^2)$, then:

$$\begin{aligned} g + b\overline{b} &= g_e(x^2) + x \cdot g_{o,\text{low}}(x^2) + \overline{x \cdot g_{o,\text{low}}(x^2)} \\ &\quad + (1 - x \cdot g_{o,\text{low}}(x^2)) \cdot \overline{(1 - x \cdot g_{o,\text{low}}(x^2))} \\ &= (1 + g_e + g_{o,\text{low}} \cdot \overline{g_{o,\text{low}}})(x^2) \end{aligned}$$

- ☺ We have projected the problem over $\mathcal{Z}_{d/2}$.
- ☺ This trick scales well with repetition.
- ☹ It incurs a growth on the coefficients' sizes...
- ☺ ... but composes nicely with gadget decomposition:
  - ➻ We write $g = g_0 + 2 \cdot g_1 + \cdots + 2^k g_k$,
  - ➻ Then we apply this trick on each g_i.

  This effectively mitigates the size growth.

A Scalable Solution

Attempt 2: Let $a\bar{a} = g$; g is self-adjoint so we can write $g = g_{\mathrm{low}} + \overline{g_{\mathrm{low}}}$. Let $b(x) = 1 - x \cdot g_{o,\mathrm{low}}(x^2)$, then:

$$\begin{aligned} g + b\bar{b} &= g_e(x^2) + x \cdot g_{o,\mathrm{low}}(x^2) + \overline{x \cdot g_{o,\mathrm{low}}(x^2)} \\ &\quad + (1 - x \cdot g_{o,\mathrm{low}}(x^2)) \cdot \overline{(1 - x \cdot g_{o,\mathrm{low}}(x^2))} \\ &= (1 + g_e + g_{o,\mathrm{low}} \cdot \overline{g_{o,\mathrm{low}}})(x^2) \end{aligned}$$

- ☺ We have projected the problem over $\mathcal{Z}_{d/2}$.
- ☺ This trick scales well with repetition.
- ☹ It incurs a growth on the coefficients' sizes...
- ☺ ... but composes nicely with gadget decomposition:
  - We write $g = g_0 + 2 \cdot g_1 + \cdots + 2^k g_k$,
  - Then we apply this trick on each g_i.

  This effectively mitigates the size growth.

Consequence: We can compute $b_1, \ldots, b_k$ in $\mathcal{Z}_d$ such that

$$a\bar{a} + \sum_i b_i \overline{b_i} = C,$$

with $k = \tilde{O}(\log \|g\|_\infty + \log d)$.

Problem 3 - Efficient Lattice Decoding

Problem: Given $\mathbf{B} \in \mathcal{Z}_d^{n \times n}$ and $\mathbf{c} \in \mathrm{Span}_{\mathcal{Q}_d}(\mathbf{B})$, compute $\mathbf{v} \in \Lambda(\mathbf{B})$ such that

$$||\mathbf{v} - \mathbf{c}|| \text{ is small.}$$

Equivalent: Given $\mathbf{B} \in \mathcal{Z}_d^{n \times n}$ and $\mathbf{t} \in \mathcal{Q}_d^n$, compute $\mathbf{z} \in \mathcal{Z}_d^n$ such that

$$||(\mathbf{z} - \mathbf{t}) \cdot \mathbf{B}|| \text{ is small.}$$

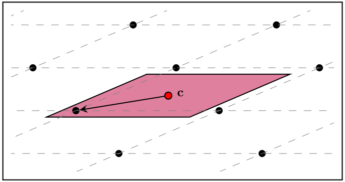
Algorithmic solutions:

- High quality, $O((nd)^2)$ operations — (Randomized) nearest plane [Bab85, GPV08]
- Lower quality, $O(n^2 d \log d)$ operations — (Randomized) round-off [Bab85, Pei10]
- High quality, $O(n^2 d \log d)$ operations — Fast Fourier orthogonalization `ia.cr/2015/1014`

Round-Off Algorithm:

1. $\mathbf{t} \leftarrow \mathbf{c} \cdot \mathbf{B}^{-1}$
2. $\mathbf{z} \leftarrow \lfloor \mathbf{t} \rceil$
3. Output $\mathbf{v} \leftarrow \mathbf{z} \cdot \mathbf{B}$

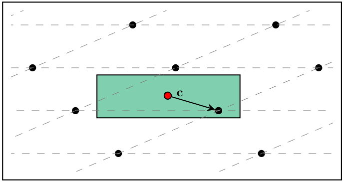
Output:



Nearest Plane Algorithm:[1]

1. $\mathbf{t} \leftarrow \mathbf{c} \cdot \mathbf{B}^{-1}$
2. For $j = n$ down to 1:
   1. $\hat{t}_j \leftarrow t_j + \sum_{i>j} (t_i - z_i) \cdot L_{i,j}$
   2. $z_j \leftarrow \lfloor \hat{t}_j \rceil$
3. Output $\mathbf{v} \leftarrow \mathbf{z} \cdot \mathbf{B}$

Output:

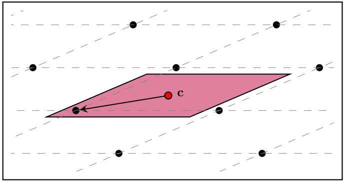


[1] Requires precomputing the Gram-Schmidt orthogonalisation (GSO) of $\mathbf{B}$: $\mathbf{B} = \mathbf{L} \cdot \tilde{\mathbf{B}}$..

Round-Off Algorithm:

1. $\mathbf{t} \leftarrow \mathbf{c} \cdot \mathbf{B}^{-1}$
2. $\mathbf{z} \leftarrow \lfloor \mathbf{t} \rceil$
3. Output $\mathbf{v} \leftarrow \mathbf{z} \cdot \mathbf{B}$

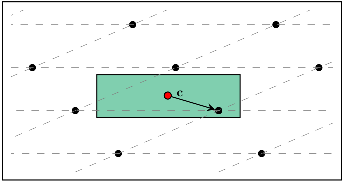
Output:



Nearest Plane Algorithm:[1]

1. $\mathbf{t} \leftarrow \mathbf{c} \cdot \mathbf{B}^{-1}$
2. For $j = n$ down to 1:
   1. $\hat{t}_j \leftarrow t_j + \sum_{i>j}(t_i - z_i) \cdot L_{i,j}$
   2. $z_j \leftarrow \lfloor \hat{t}_j \rceil$
3. Output $\mathbf{v} \leftarrow \mathbf{z} \cdot \mathbf{B}$

Output:



[1] Requires precomputing the Gram-Schmidt orthogonalisation (GSO) of **B**: $\mathbf{B} = \mathbf{L} \cdot \tilde{\mathbf{B}}$.

Tricks and Tips (1/2)

Consider the simplified case where we want this to be small:

$$(z - t) \cdot b$$

Using the ring isomorphism $\mathcal{Q}_d \cong (\mathcal{Q}_{d/2})^2$, this is equivalent to:

$$\left[\begin{array}{c|c} z_e - t_e & z_o - t_o \end{array}\right] \cdot \underbrace{\left[\begin{array}{c|c} b_e & b_o \\ \hline x b_o & b_e \end{array}\right]}_{\mathbf{B}}$$

Why this is nice:

- We can orthogonalize the second row of **B** w.r.t. to the first one:

$$\tilde{\mathbf{b}}_2 \leftarrow \mathbf{b}_2 - \underbrace{\frac{\langle \mathbf{b}_2, \mathbf{b}_1 \rangle}{\mathbf{b}_2, \mathbf{b}_1}}_{L_{2,1}} \cdot \mathbf{b}_1$$

- We can apply this "break and orthogonalize" trick recursively.
- This structured decomposition then allows a faster nearest plane algorithm.

Tricks and Tips (2/2)

Additional tricks:

- **Equivalent decomposition:**

$$\underbrace{(\mathbf{B} = \mathbf{L} \cdot \tilde{\mathbf{B}})}_{\text{GSO}} \Longleftrightarrow \underbrace{(\mathbf{B} \cdot \mathbf{B}^\star = \mathbf{L} \cdot \tilde{\mathbf{B}}\tilde{\mathbf{B}}^\star \cdot \mathbf{L}^\star)}_{\text{LDL decomposition}}$$

  The LDL decomposition is more amenable to a recursive application of our trick; this yields a complexity $O(d \log^2 d)$.

- **Working only in the FFT domain:** Discarding useless conversions further reduces the total complexity to $O(d \log d)$.

Summary (non-exhaustive)

Speed-ups in the presence of a ring:

- Most of efficient lattice-based cryptography

Speed-ups in the presence of tower of rings (**this talk**):

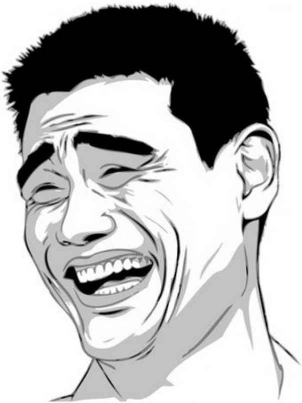
- Using ring isomorphisms: `ia.cr/2015/1014`
- Using the field norm: `ia.cr/2019/015`
- Using trace-like properties: `ia.cr/2019/230`

Exploiting automorphisms:

- Homomorphic encryption
- Zero-Knowledge proofs [dPLS18]

Key takeaway

If you cannot trivially exploit the presence of a ring...



... use its particular structure!

L Babai.
On lova´sz’ lattice reduction and the nearest lattice point problem.
In *Proceedings on STACS 85 2Nd Annual Symposium on Theoretical Aspects of Computer Science*, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

Joppe W. Bos, Kristin Lauter, Jake Loftus, and Michael Naehrig.
Improved security for a ring-based fully homomorphic encryption scheme.
In Martijn Stam, editor, *14th IMA International Conference on Cryptography and Coding*, volume 8308 of *LNCS*, pages 45–64. Springer, Heidelberg, December 2013.

Peter Campbell and Michael Groves.
Practical post-quantum hierarchical identity-based encryption.
16th IMA International Conference on Cryptography and Coding, 2017.
`http://www.qub.ac.uk/sites/CSIT/FileStore/Filetoupload,785752,en.pdf`.

Léo Ducas, Vadim Lyubashevsky, and Thomas Prest.
Efficient identity-based encryption over NTRU lattices.

In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 22–41. Springer, Heidelberg, December 2014.

Léo Ducas and Phong Q. Nguyen.
Faster Gaussian lattice sampling using lazy floating-point arithmetic.
In Wang and Sako [WS12], pages 415–432.

Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler.
Lattice-based group signatures and zero-knowledge proofs of automorphism stability.
In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 18*, pages 574–591. ACM Press, October 2018.

Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan.
Trapdoors for hard lattices and new cryptographic constructions.
In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte.
NTRUSIGN: Digital signatures using the NTRU lattice.

In Marc Joye, editor, *CT-RSA 2003*, volume 2612 of *LNCS*, pages 122–140. Springer, Heidelberg, April 2003.

Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan.
On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption.
In Howard J. Karloff and Toniann Pitassi, editors, *44th ACM STOC*, pages 1219–1234. ACM Press, May 2012.

Chris Peikert.
An efficient and parallel Gaussian sampler for lattices.
In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 80–97. Springer, Heidelberg, August 2010.

Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang.
Falcon.
Technical report, National Institute of Standards and Technology, 2017.
available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.

John M. Schanck, Andreas Hulsing, Joost Rijneveld, and Peter Schwabe.
Ntru-hrss-kem.
Technical report, National Institute of Standards and Technology, 2017.
available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Damien Stehlé and Ron Steinfeld.
Making NTRU as secure as worst-case problems over ideal lattices.
In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 27–47. Springer, Heidelberg, May 2011.

Xiaoyun Wang and Kazue Sako, editors.
ASIACRYPT 2012, volume 7658 of *LNCS*. Springer, Heidelberg, December 2012.

Zhenfei Zhang, Cong Chen, Jeffrey Hoffstein, and William Whyte.
pqntrusign.
Technical report, National Institute of Standards and Technology, 2017.
available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.