National Taiwan University, Academia Sinica, and Max Planck Institute for Security and Privacy

Algorithmic Views of Vectorized Polynomial Multipliers – NTRU

Han-Ting Chen, Yi-Hua Chung, Vincent Hwang, and Bo-Yin Yang

December 11, 2023

Goals

- Polynomial multiplications in
$$\frac{\mathbb{Z}_{2^k}[x]}{\langle x^n - 1\rangle}$$
for a prime n used in NTRU.
  - `ntruhps2048677`: $\mathbb{Z}_{2^{11}}[x]/\langle x^{677} - 1\rangle$.
  - `ntruhrss701`: $\mathbb{Z}_{2^{13}}[x]/\langle x^{701} - 1\rangle$.
- Toom-5 requiring 2^{-3}.
- Vectorization:
  - Vector-by-vector multiplications: standard vectorization.
  - Vector-by-scalar multiplications: Toeplitz matrix-vector product $\longrightarrow$ save permutations.

Toom–Cook

Toom–Cook

How to compute $\boldsymbol{ab} \in R[x]$ from $\boldsymbol{a}, \boldsymbol{b} \in R[x]_{<k}$ with Toom-k?

1. Choose $\{s_0, \dots, s_{2k-2}\} \subset \mathbb{Q} \cup \{\infty\}$
2. Apply $R[x]_{<k} \hookrightarrow R[x]/\langle \prod_i (x - s_i) \rangle \cong \prod_i R[x]/\langle x - s_i \rangle$.
3. $R[x]_{<n} \hookrightarrow R[x]/\left\langle x^{\frac{n}{k}} - y \right\rangle [y]_{<k} \hookrightarrow R[x]/\langle \boldsymbol{g} \rangle [y]_{<k}$ with $\deg(\boldsymbol{g}) \geq \frac{2n}{k} - 1$.
4. $R = \mathbb{Z}_{2^k}$:
   - We are not guaranteed to have an isomorphism over $\mathbb{Z}_{2^k}$.
   - Formally, localization of a commutative ring.
   - In practice, monomorphism suffices:
     - Identify the smallest 2^{-m} required for the correctness over $\mathbb{Q}$.
     - Compute the 2^m-multiple of the result entirely over $\mathbb{Z}_{2^{k+m}} \longrightarrow$ monomorphism.
     - Divide by $2^m \longrightarrow$ right-shift m bits.
5. NTRU with 16-bit arithmetic:
   - `ntruhps2048677`: $R = \mathbb{Z}_{2^{11}}$, we can adjoin up to 2^{-5}.
   - `ntruhrss701`: $R = \mathbb{Z}_{2^{13}}$, we can adjoin up to 2^{-3}.

Neon Vector Instruction Set

Neon Vector Instruction Set

- 32 vector registers.
  - Each vector registers holds 128-bit of data $\longrightarrow 8$ coefficients in our context.
- Arithmetic: component-wise addition/subtraction and:
  - Vector-by-vector multiplication: component-wise multiplication.
  - Vector-by-scalar multiplication: multiply a vector by a scalar (from a lane), and return a vector.
- Extending, narrowing, permutation instructions.

Neon-Optimized Toom–Cook

Neon-Optimized Toom–Cook

- 32 registers $\longrightarrow$ doubly many registers compared to existing well-studied assembly/intrinsics-optimized works (Armv7-M, AVX2).
- Existing works: Toom-2 (Karatsuba): 1; Toom-3: 2^{-1}; Toom-4: 2^{-3}.
- Toom-4 $\to$ Toom-5?
  - Register pressure: $\checkmark$
  - $\{0, \pm 1, \pm 2, \pm 3, 4, \infty\}$ vs $\underline{\{0, \pm 1, \pm 2, \pm \frac{1}{2}, 3, \infty\}}$: the former requires 2^{-4} and the later requires 2^{-3}.
- `ntruhps2048677`:
  - $R[x]_{<720} \stackrel{\text{Toom-5}}{\longrightarrow} (R[x]_{<144})^9 \stackrel{\text{Toom-3}}{\longrightarrow} (R[x]_{<48})^{45} \stackrel{\text{Toom-3}}{\longrightarrow} (R[x]_{<16})^{225} \stackrel{\text{Toom-2}}{\longrightarrow} (R[x]_{<8})^{675}$.
  - $2^{-3} \cdot 2^{-1} \cdot 2^{-1} \cdot 1 = 2^{-5}$.
  - Replace $R = \mathbb{Z}_{2^{11}}$ by $\mathbb{Z}_{2^{16}}$ in the above to adjoin 2^{-5}.

Toeplitz Matrix-Vector Product

Toeplitz Matrices

$$M = \begin{pmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \\ a_n & a_{n-1} & \cdots & a_2 & a_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m+n-3} & a_{m+n-4} & \cdots & a_{m-1} & a_{m-2} \\ a_{m+n-2} & a_{m+n-3} & \cdots & a_m & a_{m-1} \end{pmatrix}$$, for all possible i, j, $M_{i,j} = M_{i+1,j+1}$.

Denote $M = \textbf{Toeplitz}_{m \times n}(a_{m+n-2}, \ldots, a_0)$.

Toeplitz Matrix-Vector Product (TMVP, Small-Dimensional)

Compute $\begin{pmatrix} a_0 & a'_1 & a'_2 & a'_3 \\ a_1 & a_0 & a'_1 & a'_2 \\ a_2 & a_1 & a_0 & a'_1 \\ a_3 & a_2 & a_1 & a_0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}$ via vector-by-scalar multiplications.

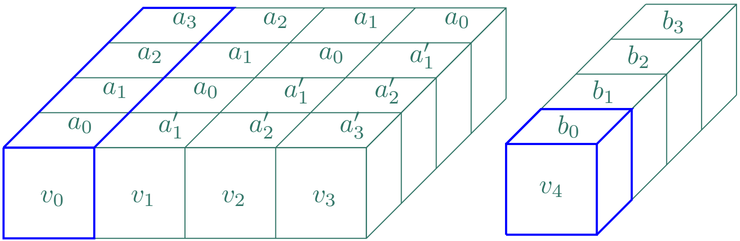

Figure: Vector-by-scalar multiplication, a degeneration of outer product.

Computing via TMVP

1. Interprate the computational problem as a TMVP.
   - Multiplying in $R[x]/\langle x^n - 1\rangle$ can be regarded as a TMVP of dimension $\geq n$.
2. Map small-dimensional TMVP to vector-by-scalar multiplications (previous slide).
3. Pick an f with $\boldsymbol{ab} = f^{-1}\left(f_k(\boldsymbol{a}) f_k(\boldsymbol{b})\right) \in R[x]$, $f_k := f|_{R[x]_{<k}}$, $f^{-1} : f\left(R[x]_{<k}\right) \to R[x]$.
4. We have **Toeplitz**$(-)\,(\boldsymbol{a}) = \mathrm{rev} \circ f_k^* \circ \left(\boldsymbol{b}' \mapsto f_k(\boldsymbol{a})\boldsymbol{b}'\right)^* \circ \left(f^{-1}\right)^*$.

- Complexity is almost the same as $\boldsymbol{ab} = f^{-1}\left(f_k(\boldsymbol{a}) f_k(\boldsymbol{b})\right)$.
- **Toeplitz**$(-)\,(\boldsymbol{a})$ decomposes into suitable TMVPs **even if f doesn't result in suitable TMVPs**.
- Suitable TMVP?
  - Example: $\begin{pmatrix} a_1 & a_0 \\ a_2 & a_1 \end{pmatrix}$.
  - Counter example: $\begin{pmatrix} a_0 & 0 \\ a_1 & a_0 \\ 0 & a_1 \end{pmatrix}, \begin{pmatrix} a_0 & a_1 \\ a_1 & a_0 + \sqrt{2}a_1 \end{pmatrix}$.
- Suppose f doesn't result in suitable TMVPs:
  - Direct translation into implementation: permutations + vector-by-vector multiplications.
  - Toeplitz: vector-by-scalar multiplications.

Comparisons of Strategies

Comparisons of Strategies

We track the subproblem sizes of algorithms used in `ntruhps2048677` works.

- [IKPC22], Toeplitz with Toom–Cook:

$$R[x]_{<720} \stackrel{4\to7}{\to} R[x]_{<180} \stackrel{3\to5}{\to} R[x]_{<60} \stackrel{3\to5}{\to} R[x]_{<20} \stackrel{2\to3}{\to} R[x]_{<10}$$

- [NG21], Toom–Cook:

$$R[x]_{<720} \stackrel{3\to5}{\to} R[x]_{<240} \stackrel{4\to7}{\to} R[x]_{<60} \stackrel{2\to3}{\to} R[x]_{<30} \stackrel{2\to3}{\to} R[x]_{<15}$$

- This work, Toeplitz with Toom–Cook:

$$R[x]_{<720} \stackrel{5\to9}{\to} R[x]_{<144} \stackrel{3\to5}{\to} R[x]_{<48} \stackrel{3\to5}{\to} R[x]_{<16} \stackrel{2\to3}{\to} R[x]_{<8}$$

For `ntruhrss701`, the Toom-3 for Toeplitz is replaced by 3-way Karatsuba (this doesn't require dividing by 2^{-k}).

Results

Polynomial Multiplications

Table: Overview of `polymuls`.

	`ntruhps2048677`	`ntruhrss701`
Implementation	Cycles	
[NG21]	58 286	70 061
`Toeplitz-TC`	26 784	31 478
`Toom-Cook`	37 318	-

- $58\,286 \rightarrow 37\,318$: Toom-4 $\rightarrow$ Toom-5 + Toom-2 $\rightarrow$ Toom-3 + memory opt.
- $37\,318 \rightarrow 26\,784$: Convert Toom–Cook into the Toeplitz form.

Polynomial Inversions

Idea for arithmetic in $\mathbb{Z}_2, \mathbb{Z}_3$: bit-level computations in batch.

Table: Performance of inversions and sorting network in NTRU.

Operation	Ref	Ours	Ref	Ours
	ntruhps2048677		ntruhrss701	
poly_Rq_inv	3 506 621	341 482	3 938 579	392 478
poly_R2_inv	2 791 906	136 776	3 175 330	140 290
poly_S3_inv	4 153 823	482 005	4 765 259	503 590
crypto_sort_int32	104 691	17 819	-	-

Scheme

Table: Overall cycles of `ntruhps2048677` and `ntruhrss701`. **K** stands for key generation, **E** stands for encapsulation, and **D** stands for decapsulation.

	`ntruhps2048677`			`ntruhrss701`		
Operation	**K**	**E**	**D**	**K**	**E**	**D**
Ref	8 245 039	227 980	331 274	9 397 305	134 737	365 558
[NG21]	7 686 272	196 526	212 265	8 599 610	87 380	221 986
`Toeplitz-TC`	1 002 187	79 213	120 208	1 076 810	59 625	142 174
`Toom-Cook`	1 127 089	88 037	146 422	-	-	-
Improvement	inv. $(\mathbb{Z}_2)$ > polymul. > sort.	sort. > polymul.	polymul.	inv. $(\mathbb{Z}_2)$ > polymul.	polymul.	polymul.

Thanks for listening

Paper (IACR ePrint): `https://eprint.iacr.org/2023/1637`

Artifact: `https://github.com/vector-polymul-ntru-ntrup/NTRU`

Reference

[IKPC22] Írem Keskinkurt Paksoy and Murat Cenk, *Faster NTRU on ARM Cortex-M4 with TMVP-based multiplication*, https://ia.cr/2022/300.

[NG21] Duc Tri Nguyen and Kris Gaj, *Fast NEON-based multiplication for lattice-based NIST post-quantum cryptography finalists*, Post-Quantum Cryptography: 12th International Workshop, PQCrypto 2021, Daejeon, South Korea, July 20–22, 2021, Proceedings, 2021, https://link.springer.com/chapter/10.1007/978-3-030-81293-5_13, pp. 234–254.

TMVP (Large-Dimensional) I

$\boldsymbol{a} \in R[x]_{<n}$, $\boldsymbol{g} = x^n - 1 \in R[x]$, $R[x]/\langle \boldsymbol{g} \rangle = R[x]_{<n}$ as modules. Suppose we have

$$\boldsymbol{b} \mapsto \boldsymbol{ab} : R[x]_{<n} \to R[x]_{<2n-1}.$$

- A straightforward way:

$$\begin{cases} R[x]/\langle \boldsymbol{g} \rangle & \to & R[x]_{<2n-1} & \to & R[x]/\langle \boldsymbol{g} \rangle \\ \boldsymbol{b} & \mapsto & \boldsymbol{ab} & \mapsto & \boldsymbol{ab} \bmod \boldsymbol{g} \end{cases}$$

- Downside: may not result in small-dimensional TMVP $\longrightarrow$ vector-by-vector + extra permutations.
- TMVP: implement $\boldsymbol{b} \mapsto \boldsymbol{ab} \bmod \boldsymbol{g}$ from $(\boldsymbol{b} \mapsto \boldsymbol{ab})^*$.

TMVP (Large-Dimensional) II

$\texttt{Expand}_{n\to n,1} = (b_i) \mapsto (b_{n-1}, \ldots, b_0, b_{n-1}, \ldots, b_1)$.

$$
\begin{tikzcd}
R[x]_{<n} \arrow[r, "\boldsymbol{b} \mapsto \boldsymbol{ab}"] & R[x]_{<2n-1} & \\
\left(R^n\right)^* \arrow[d, "\mathsf{rev} \circ \boldsymbol{c}^* \mapsto \boldsymbol{c}"'] & \left(R^{2n-1}\right)^* \arrow[l, "(\boldsymbol{b} \mapsto \boldsymbol{ab})^*"'] & \\
R^n \arrow[d, "\mathrm{id}"'] & R^{2n-1} \arrow[l, "\boldsymbol{b}' \mapsto \textbf{Toeplitz}(\boldsymbol{b}')(\boldsymbol{a})"'] \arrow[u, "\boldsymbol{c} \mapsto \boldsymbol{c}^*"'] & R^n \arrow[l, "\boldsymbol{b} \mapsto \texttt{Expand}_{n\to n,1}(\boldsymbol{b})"'] \\
R[x]/\langle \boldsymbol{g} \rangle & & R[x]/\langle \boldsymbol{g} \rangle \arrow[ll, "\boldsymbol{b} \mapsto \boldsymbol{ab} \bmod \boldsymbol{g}"'] \arrow[u, "\mathrm{id}"']
\end{tikzcd}
$$

- Generalize to $\texttt{Expand}_{n\to n',\zeta}$ (see Sections 4.3 and 4.5 of the paper).

TMVP (Large-Dimensional) III

- rev $\circ\, (\boldsymbol{b} \mapsto \boldsymbol{ab})^* = \boldsymbol{b}' \mapsto$ **Toeplitz**$(\boldsymbol{b}')(\boldsymbol{a})$
- Suppose $\exists f, \forall \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{ab} = f^{-1}\left(f_n(\boldsymbol{a}) f_n(\boldsymbol{b})\right)$, $f_n \coloneqq f_{R_{<n}[x]}$.
- If $\boldsymbol{b} \mapsto \boldsymbol{ab} = f^{-1} \circ (f_n(\boldsymbol{b}) \mapsto f_n(\boldsymbol{a}) f_n(\boldsymbol{b})) \circ f_n$ in a block matrix view, then

$$\boldsymbol{b}' \mapsto \textbf{Toeplitz}(\boldsymbol{b}')(\boldsymbol{a}) = \text{rev} \circ f_n^* \circ (f_n(\boldsymbol{b}) \mapsto f_n(\boldsymbol{a}) f_n(\boldsymbol{b}))^* \circ f^{-1*}$$

  also in a block matrix view.
- There is always such a block matrix view. Why? Each entries is a 1×1 block matrix.
- Usually,
  - f factors into a series of homomorphisms.
  - Block sizes decrease gradually (instead of dropping from n to 1 directly).