Multi-Parameter Support with NTTs for NTRU and NTRU Prime on Cortex-M4

Erdem Alkim Vincent Hwang Bo-Yin Yang

CHES 2022, Leuven, Belgium

Organization of This Talk

Contributions

Our NTT-based Polynomial Multiplications

- Compute the results in $\mathbb{Z}[x] \Longrightarrow$ reducing engineering effort
  - Compute in $\mathbb{Z}_{q'}[x]/\langle x^n - 1\rangle$
  - n: $1440, 1536, 1728$
- For $2 \nmid r$, radix-r butterfly: replace $r-1$ multiplications with $r-1$ add./sub.
  - Extend the existence of subtraction in radix-2 to other radices
  - $r = \underline{3, 5}, 17, 257, 65537,$ Fermat primes
- Good–Thomas FFT as an algebra isomorphism: $x^v \sim x^{(0)}x^{(1)}$
  - Vectorization friendly [FP07]
  - Address potential code size issues with implicit permutations
- Vector–radix FFT
  - Reduce # mul. if more than one layer in each dimension
  - Dedicated radix-$(2, 3)$ butterflies for implicit permutations

Summary of Applicability

Table 1: Overall strategies. Checked are applicable, starred are implemented.

	NTRU (n, q)			NTRU Prime (p, q)		
Conv.	$(677, 2048)$	$(701, 8192)$	$(821, 4096)$	$(653, 4621)$	$(761, 4591)$	$(857, 5167)$
Size-1440	✓*	✓*	-	✓*	-	-
Size-1536	✓*	✓*	-	✓*	✓*	-
Size-1728	✓	✓	✓*	✓	✓	✓*

Backgrounds

Polynomial Multiplications in NTRU and NTRU Prime

- "Big by small" polynomial multiplications: coeffs. of one input in $\{0, \pm 1\}$
- NTRU:
  - Rings $\mathbb{Z}_q[x]/\langle x^n - 1\rangle$, $\mathbb{Z}_q[x]/\langle x^{n-1} + \cdots + 1\rangle$, $\mathbb{Z}_3[x]/\langle x^{n-1} + \cdots + 1\rangle$
  - NTRU-HPS (`ntruhps`): $(q, n) = (2048, 509), (2048, 677), (4096, 821), (4096, 1229)$
  - NTRU-HRSS (`ntruhrss`): $(q, n) = (8192, 701), (16384, 1373)$
- NTRU Prime:
  - Rings $\mathbb{Z}_q[x]/\langle x^p - x - 1\rangle$, $\mathbb{Z}_3[x]/\langle x^p - x - 1\rangle$
  - Streamlined NTRU Prime (`sntrup`)
  - NTRU LPrime (`ntrulpr`)
  - $(q, p) = (4621, 653), (4591, 761), (5167, 857), (6343, 953), (7177, 1013), (7879, 1277)$

NTTs and FFTs

Number–Theoretic Transforms

- Ring R
- $n \perp \mathrm{char}(R)$
- $\exists$ principal n-th root of unity $\omega_n : \forall 1 \leq i < n, \sum_{0 \leq j < n} \omega_n^{ij} = 0$
- $R[x]/\langle x^n - \zeta^n \rangle \cong \prod_{i=0}^{n-1} R[x]/\langle x - \zeta\omega_n^i \rangle$
- $\boldsymbol{a}(x) \mapsto \boldsymbol{a}(\zeta\omega_n^i)_i$, invertible $\zeta \in R$

Good–Thomas FFT

- Group algebra isomorphism (explained in [Ber01], implemented in [ACC+21]):
  - Let $G \cong G_0 \times G_1$ be a group isomorphism: $R[G] \cong R[G_0] \otimes R[G_1]$
  - $q_0 \perp q_1 \longrightarrow \mathbb{Z}_{q_0 q_1} \cong \mathbb{Z}_{q_0} \times \mathbb{Z}_{q_1}$: $\frac{R[x]}{\langle x^{q_0 q_1}-1\rangle} \cong \frac{R[x]}{\langle (x^{(0)})^{q_0}-1\rangle} \otimes \frac{R[x]}{\langle (x^{(1)})^{q_1}-1\rangle}$
- Algebra isomorphism (already implied in [Goo58, FP07]):
  - $\left\{ \boldsymbol{a}(x) \mapsto \boldsymbol{a}(\omega_{q_0 q_1}^{i})_i \right\} \cong \left\{ \left(\boldsymbol{a}(x^{(0)}) \mapsto \boldsymbol{a}(\omega_{q_0}^{i_0})_{i_0} \right) \otimes \left(\boldsymbol{a}(x^{(1)}) \mapsto \boldsymbol{a}(\omega_{q_1}^{i_1})_{i_1} \right) \right\}$
  - $\frac{R[x]}{\langle x^{vq_0q_1}-1\rangle} \cong \frac{R[x,y]}{\langle x^v-y, y^{q_0q_1}-1\rangle} \cong \frac{R[x,y,u,w]}{\langle x^v-y, y-uw, u^{q_0}-1, w^{q_1}-1\rangle} \cong \prod_{i_0,i_1} \frac{R[x,y,u,w]}{\left\langle x^v-y, y-uw, u-\omega_{q_0}^{i_0}, w-\omega_{q_1}^{i_1} \right\rangle}$
    $= \prod_{i_0,i_1} \frac{R[x,y]}{\left\langle x^v-y, y-\omega_{q_0}^{i_0}\omega_{q_1}^{i_1} \right\rangle} = \prod_{i_0,i_1} \frac{R[x]}{\left\langle x^v-\omega_{q_0}^{i_0}\omega_{q_1}^{i_1} \right\rangle} = \prod_{i_0,i_1} \frac{R[x]}{\left\langle x^v-\omega_{q_0q_1}^{e_0 i_0+e_1 i_1} \right\rangle}$

Vector–Radix FFT

- For well-defined f_0, f_1, g_0, g_1, $(f_0 \circ f_1) \otimes (g_0 \circ g_1) = (f_0 \otimes g_0) \circ (f_1 \otimes g_1)$
- $(f_0 \circ \cdots \circ f_{d-1}) \otimes (g_0 \circ \cdots \circ g_{d-1}) = (f_0 \otimes g_0) \circ \cdots \circ (f_{d-1} \otimes g_{d-1})$
- $\texttt{NTT}^{(0)} := \texttt{add} \circ \texttt{mul} \circ \cdots \texttt{add} \circ \texttt{mul}$
- $\texttt{NTT}^{(1)} := \texttt{add} \circ \texttt{mul} \circ \cdots \texttt{add} \circ \texttt{mul}$
- $\texttt{NTT}^{(0)} \otimes \texttt{NTT}^{(1)} = (\texttt{add} \otimes \texttt{add}) \circ \underline{(\texttt{mul} \otimes \texttt{mul})} \circ \cdots \circ (\texttt{add} \otimes \texttt{add}) \circ \underline{(\texttt{mul} \otimes \texttt{mul})}$
- $\left(x^{(1)} \mapsto \zeta_1\right) \otimes \left(x^{(0)} \mapsto \zeta_0\right) = \left(x^{(0)}\right)^{i_0} \left(x^{(1)}\right)^{i_1} \mapsto \zeta_0^{i_0} \zeta_1^{i_1}$
  - $2q_0q_1 - q_0 - q_1$ multiplications $\Longrightarrow q_0q_1 - 1$ multiplications
- Radix-(r_0, r_1): radix-r_0 butterfly $\otimes$ radix-r_1 butterfly

Improved Näive Butterflies

Näive Butterflies

- $\begin{pmatrix} \boldsymbol{c}(\psi) \\ \boldsymbol{c}(\psi\omega_3) \\ \boldsymbol{c}(\psi\omega_3^2) \end{pmatrix} = \begin{pmatrix} c_0 + \underline{\psi c_1 + \psi^2 c_2} \\ c_0 + \underline{\psi\omega_3 c_1 + \psi^2\omega_3^2 c_2} \\ c_0 + \underline{\psi\omega_3^2 c_1 + \psi^2\omega_3 c_2} \end{pmatrix}$
- `smull`, `smlal` followed by Montgomery reduction (`mul`, `smlal`)
- $\psi \neq 1 \implies$ 15 cycles $(5+5+5)$; $\psi = 1 \implies$ 12 cycles $(2+5+5)$

Algorithm 1 Näive butterflies

1: `smull t1, c0', c1,` ψ ▷ The last operand is $\psi\omega_3$ for `c1'`,$\psi\omega_3^2$ for `c2'`
2: `smlal t1, c0', c2,` ψ^2 ▷ The last operand is $\psi^2\omega_3^2$ for `c1'`,$\psi^2\omega_3$ for `c2'`
3: `mul t0, t1,` q'
4: `smlal t1, c0', t0,` q ▷ `c0'` $= \psi c_1 + \psi^2 c_2$. If $\psi = 1$, `c0'` $= c_1 + c_2$ with `add`
5: ▷ Compute $\boldsymbol{c}(\psi) =$ `c0'` $+ c_0$, $\boldsymbol{c}(\psi\omega_3) =$ `c1'` $+ c_0$, $\boldsymbol{c}(\psi\omega_3^2) =$ `c1'` $+ c_0$

Improved Näive Butterflies i

Algorithm 2 Improved näive butterflies

1: ... ▷ c0' $= \psi c_1 + \psi^2 c_2$. If $\psi = 1$, c0' $= c_1 + c_2$ with add
2: `smull t1, c1', c1,` $\psi\omega_3$
3: `smlal t1, c1', c2,` $\psi^2\omega_3^2$
4: `mul t2, t1,` q'
5: `smlal t1, c1', t2,` q ▷ c1' $= \psi\omega_3 c_1 + \psi^2\omega_3^2 c_2$
6: `add c2', c1', c0` ▷ c2' $= (\psi c_1 + \psi^2 c_2) + (\psi\omega_3 c_1 + \psi^2\omega_3^2 c_2) = -\psi\omega_3^2 c_1 - \psi^2\omega_3 c_2$
7: `sub c2, c0, c2'` ▷ c2 $= \boldsymbol{c}(\psi\omega_3^2)$
8: `add c1, c0, c1'` ▷ c1 $= \boldsymbol{c}(\psi\omega_3)$
9: `add c0, c0, c0'` ▷ c0 $= \boldsymbol{c}(\psi)$

Improved Näive Butterflies ii

Generalize to radix-r butterflies.

- $\sum_{i=0}^{r-1} \boldsymbol{c}(\psi\omega_r^i) = rc_0$
- For a j, $\boldsymbol{c}(\psi\omega_r^j) = rc_0 - \sum_{i=0, i\neq j}^{r-1} \boldsymbol{c}(\psi\omega_r^i) = c_0 - \sum_{i=0, i\neq j}^{r-1} \left(\boldsymbol{c}(\psi\omega_r^i) - c_0\right)$
  - Compute $\boldsymbol{c}(\psi\omega_r^i) - c_0 = \sum_{j=1}^{r-1} c_j \psi^j \omega_r^{ij}$ as usual
  - Compute $\boldsymbol{c}(\psi\omega_r^j) = c_0 - \sum_{i=0, i\neq j}^{r-1} \left(\boldsymbol{c}(\psi\omega_r^i) - c_0\right)$ with $r-1$ additions/subtractions
- r needs not to be odd, but odd numbers require more studies

Improved Näive Butterflies iii

Let r be an odd and $\psi = 1$ (the cyclic case). Many ways for $\boldsymbol{c}(x) \mapsto \boldsymbol{c}(\omega_r^i)_i$.

- Focus on prime $r = 2^{2^t} + 1$
- Fermat primes $3, 5, 17, 257, 65537$
- Radix-3 butterflies are improved
- Radix-5 butterflies are believed to be improved
- Radix-$2^{2^{\{2,3,4\}}} + 1$ butterflies are probably not improved

More on Good–Thomas FFT

Dedicated Butterflies for Implicit Permutations

- $R[x]/\langle x^{24}-1\rangle \cong \prod_{i_0,i_1} R[x,u,w]\Big/\left\langle x-uw, u-\omega_3^{i_0}, w^4-\omega_2^{i_1}\right\rangle$ or $\prod_{i'} R[x,u,w]\Big/\left\langle x-uw, u^3-1, w-\omega_8^{i'}\right\rangle$?
- At most 6 "dedicated" radix-$(3,2)$ butterflies
- Better than "dedicated" 3-layer-radix-2 butterflies [ACC+21]
- We save more because
  - Half of the entries are zeros: more saving with radix-3
  - There are more follow up radix-2 butterflies computing $(a,b)\mapsto(a+b,a-b)$

Potential Code Size Issues with Implicit Permutations

- Assume dedicated radix-$(3, 2)$ at the beginning
- Size-2^{k_0} $\otimes$ size-3^{k_1} cyclic NTTs where $3^{k_1} < 2^{k_0-1}$
  - $R[x] \Big/ \left\langle x^{2^{k_0}3^{k_1}} - 1 \right\rangle \cong \prod_{i_0,i_1} R[x, u, w] \Big/ \left\langle x - uw, u^{2^{k_0-1}} - \omega_2^{i_0}, w^{3^{k_1-1}} - \omega_3^{i_1} \right\rangle$
- A loop consisting of 3^{2k_1-1} dedicated radix-$(3, 2)$ butterflies
- Code sizes
  - $1440 = 160 \cdot 9, 3^{2k_1-1} = 27$, compact code size
  - $1536 = 512 \cdot 3, 3^{2k_1-1} = 3$, compact code size
  - $1728 = 64 \cdot 27, 3^{2k_1-1} = 243$, large code size

Our Resolution

- q_0: power of 2, q_1: power of 3 with $q_0 < \frac{q_0}{2}$
- $\tilde{q}$: how incomplete Cooley–Tukey is
- v: how incomplete Good–Thomas is
- At most one of $\tilde{q}$, v is greater than 1
- Consider $R[x]\Big/\left\langle x^{q_0\tilde{q}q_1 v} - 1\right\rangle \cong \prod_{i_0,i_1} R[x,u,w]\Big/\left\langle x^v - uw, u^{\tilde{q}} - \omega_{q_0}^{i_0}, w - \omega_{q_1}^{i_1}\right\rangle$
  - 1440 : $(q_0, \tilde{q}, q_1, v) = (32, 5, 9, 1)$
  - 1536 : $(q_0, \tilde{q}, q_1, v) = (128, 4, 3, 1)$
  - 1728 : $(q_0, \tilde{q}, q_1, v) = (64, 1, 9, 3)$

Results

Polynomial Multiplications i

Figure 1: Overall performance of polynomial multiplications.

NTRU				
(n, q)	Convolution	This work	[CHK+21]	[IKPC22]
(677, 2048)	Size-677	−/−	−/−	144k/−
	Size-1440	140k/143k	−/−	−/−
	Size-1536	147k/149k	156k/−	−/−
(701, 8192)	Size-701	−/−	−/−	144k/−
	Size-1440	141k/143k	−/−	−/−
	Size-1536	148k/150k	156k/−	−/−
(821, 4096)	Size-821	−/−	−/−	193k/−
	Size-1728	178k/182k	199k/−	−/−

NTRU Prime				
(p, q)	Convolution	This work	[ACC+21]	[Che21][1]
(653, 4621)	Size-1320	−/−	−/−	120k/−
	Size-1440	142k/147k	−/−	−/−
(761, 4591)	Size-1530	−/−	152k/−	142k/−
	Size-1536	151k/153k	159k/−	−/−
	Size-1620	−/−	185k/−	−/−
(857, 5167)	Size-1722	−/−	−/−	203k/−
	Size-1728	182k/186k	−/−	−/−

Polynomial Multiplications ii

Table 2: Detailed numbers of polynomial multiplications for NTRU.

NTRU							
(n, q)	Size	`polymul`	`NTT`	`NTT_small`	`basemul`	`iNTT`	`final_map`
(677, 2048)	1440	140 444	34 102	33 241	27 690	36 756	8 835
		143 016	34 963	34 093	27 825	37 214	9 208
(677, 2048)	1536	147 126	37 485	36 573	23 322	41 437	8 489
		149 174	38 076	37 139	23 506	42 001	8 717
(701, 8192)	1440	140 577	34 102	33 241	27 690	36 756	8 968
		143 239	34 957	34 087	27 819	37 208	9 431
(701, 8192)	1536	147 670	37 485	36 573	23 322	41 437	9 033
		149 771	38 076	37 139	23 506	42 001	9 314
(821, 4096)	1728	181 534	48 629	47 627	21 848	53 098	10 512
		186 197	49 480	48 507	22 349	55 569	10 564

Polynomial Multiplications iii

Table 3: Detailed numbers of polynomial multiplications for NTRU Prime.

NTRU Prime							
(p, q)	Size	`polymul`	`NTT`	`NTT_small`	`basemul`	`iNTT`	`final_map`
(653, 4621)	1440	142 244	34 104	33 244	27 690	36 756	10 629
		146 665	34 992	34 095	27 813	37 214	12 823
(761, 4591)	1536	151 374	37 487	36 573	23 322	41 435	12 739
		153 299	38 069	37 138	23 510	42 001	12 861
(857, 5167)	1728	184 714	48 629	47 623	21 848	53 099	13 695
		189 523	49 483	48 499	22 336	55 720	13 743

NTRU Results

- Key generation from [Li21]
- NTRU–HPS: `crypto_sort` from NTRU Prime for **K** and **E**

Table 4: Overall performance of NTRU. **K** = key generation, **E** = encryption, **D** = decryption.

	ntruhps2048677			ntruhrss701			ntruhps4096821		
	K	**E**	**D**	**K**	**E**	**D**	**K**	**E**	**D**
[CHK+21]	143 725k	821k	818k	153 403k	377k	871k	207 495k	1 027k	1 030k
[IKPC22]	142 378k	816k	729k	153 479k	369k	787k	212 377k	1 026k	914k
[Li21][1]	4 625k	820k	812k	4 233k	376k	868k	6 116k	1 027k	1 031k
This work	3 912k	525k	718k	3 822k	361k	778k	5 217k	654k	908k

NTRU Prime Results

- [ACC+21]: secrete-dependent table lookup AES

Table 5: Overall performance of NTRU Prime.

	ntrulpr653			ntrulpr761			ntrulpr857		
	K	**E**	**D**	**K**	**E**	**D**	**K**	**E**	**D**
[ACC+21][2]	-	-	-	731k	1 102k	1 200k	-	-	-
[Che21]	678k	1 158k	1 233k	727k	1 312k	1 394k	-	-	-
This work	669k	1 131k	1 231k	710k	1 266k	1 365k	886k	1 465k	1 596k
	sntrup653			sntrup761			sntrup857		
	K	**E**	**D**	**K**	**E**	**D**	**K**	**E**	**D**
[ACC+21][2]	-	-	-	10 778k	694k	572k	-	-	-
[Che21]	6 715k	632k	487k	7 951k	684k	538k	-	-	-
This work	6 623k	621k	527k	7 937k	666k	563k	10 192k	812k	685k

Future Works

- Vectorization of Good–Thomas with $x^v \sim uw, v > 1$
  - Implemented in [FP07] (SSE) using program generator Spiral
  - Recently, NTT-based RSA-4096 in [BHK+22] (MVE)
  - How about Neon, AVX2, AVX512?
- In [BBCT21] for NTRU Prime, radix-2 Schönhage for "big by big" polynomial multiplication because of vectorization

  Q1 What is the role of the existing principal 3rd root of unity in $\mathbb{Z}_{4591}$?

  Q2 How to combine vectorization-friendly Good–Thomas and Schönhage?

Thank you for your attention

Reference i

Erdem Alkim, Dean Yun-Li Cheng, Chi-Ming Marvin Chung, Hülya Evkan, Leo Wei-Lun Huang, Vincent Hwang, Ching-Lin Trista Li, Ruben Niederhagen, Cheng-Jhih Shih, Julian Wälde, and Bo-Yin Yang.
Polynomial Multiplication in NTRU Prime Comparison of Optimization Strategies on Cortex-M4.
IACR Transactions on Cryptographic Hardware and Embedded Systems, 2021(1):217–238, 2021.
https://tches.iacr.org/index.php/TCHES/article/view/8733.

Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, and Nicola Tuveri.
OpenSSLNTRU: Faster post-quantum TLS key exchange.
arXiv preprint arXiv:2106.08759, 2021.

Reference ii

Daniel J. Bernstein.
Multidigit multiplication for mathematicians.
2001.

Hanno Becker, Vincent Hwang, Matthias J. Kannwischer, Lorenz Panny, and Bo-Yin Yang.
Efficient Multiplication of Somewhat Small Integers using Number–Theoretic Transforms.
Cryptology ePrint Archive, 2022.
https://eprint.iacr.org/2022/439.

Reference iii

Yun-Li Cheng.
Number Theoretic Transform for Polynomial Multiplication in Lattice-based Cryptography on ARM Processors.
Master's thesis, 2021.
https://github.com/dean3154/ntrup_m4.

Chi-Ming Marvin Chung, Vincent Hwang, Matthias J. Kannwischer, Gregor Seiler, Cheng-Jhih Shih, and Bo-Yin Yang.
NTT Multiplication for NTT-unfriendly Rings New Speed Records for Saber and NTRU on Cortex-M4 and AVX2.
IACR Transactions on Cryptographic Hardware and Embedded Systems, 2021(2):159–188, 2021.

Reference iv

https://tches.iacr.org/index.php/TCHES/article/view/8791.

Franz Franchetti and Markus Puschel.
SIMD Vectorization of Non-Two-Power Sized FFTs.
In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 2, 2007.

I. J. Good.
The Interaction Algorithm and Practical Fourier Analysis.
Journal of the Royal Statistical Society: Series B (Methodological), 20(2):361–372, 1958.

Reference

Írem Keskinkurt Paksoy and Murat Cenk.
Faster NTRU on ARM Cortex-M4 with TMVP-based multiplication.
2022.
https://eprint.iacr.org/2022/300.

Ching-Lin Li.
Implementation of Polynomial Modular Inversion in Lattice-based cryptography on ARM.
Master's thesis, 2021.
https://github.com/trista5658321/polyinv-m4.

Reference vi

Charles M. Rader.
Discrete fourier transforms when the number of data samples is prime.
Proceedings of the IEEE, 56(6):1107–1108, 1968.

Shmuel Winograd.
On Computing the Discrete Fourier Transform.
Mathematics of computation, 32(141):175–199, 1978.

A Series of Reductions to Fermat-Prime-Size Butterflies

For $\boldsymbol{a}(x) \mapsto \left(\boldsymbol{a}(\omega_r^i)\right)_i$ with an odd r, ways for $\boldsymbol{c}(x) \mapsto \boldsymbol{c}(\omega_r^i)_i$:

- Näive size-r butterfly
- $\exists q_0 \perp q_1, r = q_0 q_1$: Good–Thomas
- Prime power $r = p^k$: Winograd's $\Longrightarrow$ size-$p^{k-1}(p-1)$ convolution [Win78]
  - $k > 1$: $p - 1 \perp p \Longrightarrow$ Good–Thomas
  - $k = 1$: Rader's $\Longrightarrow$ size-$(p-1)$ convolution [Rad68]
- Assume $r = p$, $p - 1$ is even
  - $\exists$ odd $q_0 | p - 1$: Good–Thomas
  - $p - 1 = 2^h \Longrightarrow p = F_t := 2^{2^t} + 1$

Size-1728 Convolution

$$
\begin{aligned}
R[x]/\left\langle x^{1728}-1\right\rangle &\cong \prod_{i_{u,0}=0}^{2}\prod_{i_{w,0}=0}^{1} R[x,u,w]\Big/\left\langle x^3-uw,\, u^3-\omega_3^{i_{u,0}},\, w^{32}-\omega_2^{i_{w,0}}\right\rangle \\
&\cong \prod_{i_{u,0},i_{u,1}=0}^{2}\prod_{i_{w,0},i_{w,1}=0}^{1} R[x,u,w]\Big/\left\langle x^3-uw,\, u-\omega_9^{i_{u,0}+3i_{u,1}},\, w^{16}-\omega_4^{i_{w,0}+2i_{w,1}}\right\rangle \\
&\cong \prod_{i_{u,0},i_{u,1}=0}^{2}\prod_{i_{w,0},\ldots,i_{w,5}=0}^{1} R[x,u,w]\Big/\left\langle x^3-uw,\, u-\omega_9^{i_{u,0}+3i_{u,1}},\, w-\omega_{64}^{\sum_{j=0}^{5}2^j i_{w,j}}\right\rangle \\
&= \prod_{i_{u,0},i_{u,1}=0}^{2}\prod_{i_{w,0},\ldots,i_{w,5}=0}^{1} R[x]\Big/\left\langle x^3-\omega_9^{i_{u,0}+3i_{u,1}}\omega_{64}^{\sum_{j=0}^{5}2^j i_{w,j}}\right\rangle
\end{aligned}
$$

Size-1536 Convolution

$$
\begin{aligned}
R[x]/\left\langle x^{1536}-1\right\rangle &\cong \prod_{i_{u,0}=0}^{2}\prod_{i_{w,0}=0}^{1} R[x,u,w]\Big/\left\langle x-uw,\, u-\omega_3^{i_{u,0}},\, w^{256}-\omega_2^{i_{w,0}}\right\rangle \\
&\cong \prod_{i_{u,0}=0}^{2}\prod_{i_{w,0},\ldots,i_{w,3}=0}^{1} R[x,u,w]\Big/\left\langle x-uw,\, u-\omega_3^{i_{u,0}},\, w^{32}-\omega_{16}^{\sum_{j=0}^{3}2^j i_{w,j}}\right\rangle \\
&\cong \prod_{i_{u,0}=0}^{2}\prod_{i_{w,0},\ldots,i_{w,6}=0}^{1} R[x,u,w]\Big/\left\langle x-uw,\, u-\omega_3^{i_{u,0}},\, w^{4}-\omega_{128}^{\sum_{j=0}^{6}2^j i_{w,j}}\right\rangle
\end{aligned}
$$

Size-1440 Convolution

$$
\begin{aligned}
R[x]/\left\langle x^{1440}-1\right\rangle &\cong \prod_{i_{u,0}=0}^{2}\prod_{i_{w,0}=0}^{1} R[x,u,w]\Big/\left\langle x-uw,\, u^3-\omega_3^{i_{u,0}},\, w^{80}-\omega_2^{i_{w,0}}\right\rangle \\
&\cong \prod_{i_{u,0},i_{u,1}=0}^{2}\prod_{i_{w,0},i_{w,1}=0}^{1} R[x,u,w]\Big/\left\langle x-uw,\, u-\omega_9^{i_{u,0}+3i_{u,1}},\, w^{40}-\omega_4^{i_{w,0}+2i_{w,1}}\right\rangle \\
&\cong \prod_{i_{u,0},i_{u,1}=0}^{2}\prod_{i_{w,0},\ldots,i_{w,4}=0}^{1} R[x,u,w]\Big/\left\langle x-uw,\, u-\omega_9^{i_{u,0}+3i_{u,1}},\, w^{5}-\omega_{32}^{\sum_{j=0}^{4}2^j i_{w,j}}\right\rangle
\end{aligned}
$$