MAX PLANCK INSTITUTE
FOR SECURITY AND PRIVACY

NTT Multiplication for NTT-unfriendly Rings

New Speed Records for Saber and NTRU on Cortex-M4 and AVX2

Chi-Ming Marvin Chung Vincent Hwang Matthias J. Kannwischer Gregor Seiler Cheng-Jhih Shih Bo-Yin Yang

CHES 2021 – September 17, 2021

Number-theoretic Transforms

Number-theoretic Transforms

Suppose n is invertible.

Given an invertible $\zeta \in R$ and a principle n-th root of unity ω ($\forall k, n \not| k \longrightarrow \sum_{i=0}^{n-1} \omega^{ik} = 0$), we have

$$\texttt{NTT} : R[x]/\langle x^n - \zeta^n \rangle \cong \prod_{i=0}^{n-1} R[x]/\langle x - \zeta\omega^i \rangle.$$

- Cyclic: $\zeta^n = 1$
- Negacyclic: $\zeta^n = -1$
- Cooley–Tukey: $\zeta^n = (\omega^j)^n$, the n-th power of a power of ω

As an isomorphism, we have

$$\begin{cases} \texttt{NTT}(\boldsymbol{a}(x)\boldsymbol{b}(x)) = \texttt{NTT}(\boldsymbol{a}(x))\texttt{NTT}(\boldsymbol{b}(x)) & \iff \boldsymbol{a}(x)\boldsymbol{b}(x) = \texttt{NTT}^{-1}(\texttt{NTT}(\boldsymbol{a}(x))\texttt{NTT}(\boldsymbol{b}(x))) \\ \texttt{NTT}(\boldsymbol{a}(x) + \boldsymbol{b}(x)) = \texttt{NTT}(\boldsymbol{a}(x)) + \texttt{NTT}(\boldsymbol{b}(x)) & \iff \boldsymbol{a}(x) + \boldsymbol{b}(x) = \texttt{NTT}^{-1}(\texttt{NTT}(\boldsymbol{a}(x)) + \texttt{NTT}(\boldsymbol{b}(x))) \end{cases}$$

Fast Fourier Transforms

- Let ω be a principal $n_0 n_1$-th root of unity.
  - Cooley–Tukey:

$$R[x]/\langle x^{n_0 n_1} - 1\rangle \cong \prod_{i=0}^{n_0-1} R[x]/\langle x^{n_1} - \omega^{n_1 i}\rangle \cong \prod_{i=0}^{n_0-1}\prod_{j=0}^{n_1-1} R[x]/\langle x - \omega^{i+n_0 j}\rangle$$

  - Gentleman–Sande:

$$R[x]/\langle x^{n_0 n_1} - 1\rangle \cong \prod_{i=0}^{n_0-1} R[x]/\langle x^{n_1} - \omega^{n_1 i}\rangle \cong \prod_{i=0}^{n_0-1} R[x]/\langle x^{n_1} - 1\rangle \cong \prod_{i=0}^{n_0-1}\prod_{j=0}^{n_1-1} R[x]/\langle x - \omega^{n_0 j}\rangle$$

- For $q_0 \perp q_1$, Good's trick:

$$R[x]/\langle x^{q_0 q_1} - 1\rangle \cong \left(R[z]/\langle z^{q_1} - 1\rangle\right)[y]/\langle y^{q_0} - 1\rangle$$

NTTs: Montgomery Multiplications

- Cortex-M4: `smull`, `smlal` $\Longrightarrow$ 32-bit arithmetic
- AVX2: `vpmulhw` $\Longrightarrow$ 16-bit arithmetic

Algorithm 1 32-bit Montgomery multiplication on Cortex-M4

Input: $(\mathtt{c0}, \mathtt{c1}) = (a, b)$

Output: $\mathtt{c0} \equiv ab2^{-32} \pmod{q}$

1: `smull tmp0, c0, c0, c1`
2: `mul tmp1, tmp0,` $(-q^{-1} \bmod {}^{\pm}\mathtt{R})$
3: `smlal tmp0, c0, tmp1,` q

Algorithm 2 16-bit Montgomery multiplication with AVX2

Require: $a \in [-2^{15}, 2^{15})$, $b \in [-\frac{q-1}{2}, \frac{q-1}{2}]$, $b' = bq^{-1} \bmod 2^{16}$

Ensure: $r \equiv ab2^{-16} \pmod{q}$

1: $t_1 \leftarrow \lfloor \frac{ab}{2^{16}} \rfloor$ ▷ signed high product
2: $t_0 \leftarrow ab' \bmod 2^{16}$ ▷ signed low product
3: $t_0 \leftarrow \lfloor \frac{t_0 q}{2^{16}} \rfloor$ ▷ signed high product
4: $r \leftarrow (t_1 - t_0) \bmod 2^{16}$

Saber

Saber

- $q = 8192$
- $R_q = \mathbb{Z}_q[x]/\langle x^{256} + 1 \rangle$
- Parameter (l, μ) varies for security levels:
  - `Lightsaber`: $(l, \mu) = (2, 10)$
  - `Saber`: $(l, \mu) = (3, 8)$
  - `Firesaber`: $(l, \mu) = (4, 6)$
- Compute $A^T \cdot s$ and $A \cdot s'$ where
  - $A \in R_q^{l \times l}$
  - $s, s' \in R_q^l$, coefficients are in $[-\frac{\mu}{2}, \frac{\mu}{2}]$, small

How NTT-friendly/unfriendly Saber is?

- Polynomial modulus: $x^{256} + 1$, awesome!
- Coefficient ring: $\mathbb{Z}_{8192}$, unfriendly
  - ~~Solution: choose ...~~
  - Hold on.
- Recall if NTT is defined correctly, then

$$\boldsymbol{a}(x)\boldsymbol{b}(x) + \boldsymbol{c}(x)\boldsymbol{d}(x) = \texttt{NTT}^{-1}(\texttt{NTT}(\boldsymbol{a}(x))\texttt{NTT}(\boldsymbol{b}(x)) + \texttt{NTT}(\boldsymbol{c}(x))\texttt{NTT}(\boldsymbol{d}(x)))$$

$\Longrightarrow$

$$A^T \cdot s = \texttt{NTT}^{-1}(\texttt{NTT}(A^T)\texttt{NTT}(s))$$

NTTs for $A^T \cdot s$

- We compute $A^T \cdot s$ as if $\mathbb{Z}$ is the coefficient ring
- Bounding the maximum value of the result:
  - Max: $2 \cdot 256 \cdot \frac{8192}{2} \cdot \frac{\mu}{2} \cdot l = 2^{20} \cdot \mu \cdot l$
  - Cortex-M4: choose a 32-bit prime $q' > 2^{20} \cdot \mu \cdot l$
  - AVX2: choose two 16-bit primes p_0, p_1 with $p_0 p_1 > 2^{20} \cdot \mu \cdot l$
- Compute $A^T \cdot s$ as
  - Cortex-M4: $\mathtt{NTT}^{-1}\left(\mathtt{NTT}(A^T)\mathtt{NTT}(s)\right)$ in $\mathbb{Z}_{q'}$
  - AVX2: $\mathtt{NTT}^{-1}\left(\mathtt{NTT}(A^T)\mathtt{NTT}(s)\right)$ in $\mathbb{Z}_{p_0}$ and $\mathbb{Z}_{p_1}$, and CRT at the end
- $l + l^2$ NTTs: $\mathtt{NTT}(s)$ and $\mathtt{NTT}(A^T)$
- l^2 "point multiplications"
- l $\mathtt{NTT}^{-1}$: $\mathtt{NTT}^{-1}\left(\mathtt{NTT}(A^T)\mathtt{NTT}(s)\right)$

NTTs for Saber on Cortex-M4

- Incomplete NTTs giving 4-coefficient polynomials
- Long multiplication with accumulation: `smlal`
  - Let $\mathbf{p}_i, \mathbf{q}_i$ be 4-coefficient polynomials
  - $\star$ is multiplication in $\pmod{x^4 - \zeta}$
  - For $\mathbf{h} = \sum_i \mathbf{p}_i \star \mathbf{q}_i$, $[x^0]\mathbf{h} = \sum_i \left(\mathbf{p}_{i0}\mathbf{q}_{i0} + \zeta(\mathbf{p}_{i1}\mathbf{q}_{i3} + \mathbf{p}_{i2}\mathbf{q}_{i2} + \mathbf{p}_{i3}\mathbf{q}_{i1})\right)$
  - We can
    - `montgomeryR` for each of $\mathbf{p}_{i0}\mathbf{q}_{i0} + \zeta(\mathbf{p}_{i1}\mathbf{q}_{i3} + \mathbf{p}_{i2}\mathbf{q}_{i2} + \mathbf{p}_{i3}\mathbf{q}_{i1})$, or
    - `montgomeryR` for $[x^0]\mathbf{h}$

NTRU

NTRU

- $\mathbb{Z}_q[x]/\langle x^n - 1\rangle$
- Parameter (q, n) varies for security levels:
  - `ntruhps2048509`: $(q, n) = (2048, 509)$
  - `ntruhps2048677`: $(q, n) = (2048, 677)$
  - `ntruhrss701`: $(q, n) = (8192, 701)$
  - `ntruhps4096821`: $(q, n) = (4096, 821)$
  - `ntruhps40961229`: $(q, n) = (4096, 1229)$
  - `ntruhrss1373`: $(q, n) = (16384, 1373)$
- One of the multiplicands is ternary, i.e. coefficients are in $\{\pm 1, 0\}$

NTTs for NTRU on Cortex-M4

Parameter sets	$\texttt{NTT}_N$	q'	Strategy
ntruhps4096821	$1728 = 9 \cdot 64 \cdot 3$	3365569	Mixed-radix (CT+GS)
ntruhrss701	$1536 = 512 \cdot 3$	5747201	Good's (CT+CT)
ntruhps2048677	$1536 = 512 \cdot 3$	1389569	Good's (CT+CT)
ntruhps2048509	$1024 = 256 \cdot 4$	1043969	Radix-2 (CT+GS)

In NTRU Prime, primes p, q give the field $\mathbb{Z}_q[x]/\langle x^p - x - 1\rangle$. We compare $(q, p) = (4591, 761)$ in NTRU Prime with $(q, n) = (2048, 677)$ and $(8192, 701)$ in NTRU.

- $\left\langle x^{\texttt{NTT}_N} - 1\right\rangle \to \langle x^n - 1\rangle$ is faster than $\left\langle x^{\texttt{NTT}_N} - 1\right\rangle \to \langle x^p - x - 1\rangle$. Excluding head and tail cases,
  - $x^n - \underline{1}$: 1 add; $q' > n \cdot q$
  - $x^p - \underline{(x+1)}$: 2 adds; $q' > n \cdot (2p-1)$
- $\mathbb{Z}_{q'} \to \mathbb{Z}_{\{2048,8192\}}$ is faster than $\mathbb{Z}_{q'} \to \mathbb{Z}_{4591}$:
  - pkhbt before $\mathbb{Z}_{q'} \to \mathbb{Z}_{\{2048,8192\}}$: 0.5 cycles on average with the and instruction
  - pkhbt after $\underline{\mathbb{Z}_{q'} \to \mathbb{Z}_{4591}}$: 2 cycles on average with Barrett reduction
- On average, we save 2.5 cycles for each coefficient.
- Polynomials in NTRU are shorter than in NTRU Prime $\implies > 2.5 * 701 = 1752.5$ cycles of reduction

Implementation Considerations with AVX2

NTTs with AVX2

- Divided difference form of CRT:
  - Solve $|u| < P/2 = \prod_{i=1}^{s} p_i/2$ from $u \equiv u_i \pmod{p_i}$, $i = 1 \dots s$, $|u_i| < p_i/2$
  - Let $m_i := (p_1 \cdots p_{i-1})^{-1} \bmod^{\pm} p_i$.
  -

$$\begin{cases} y_1 = u_1 \\ y_2 = y_1 + ((u_2 - y_1)m_2 \bmod^{\pm} p_2)\, p_1 \\ y_3 = y_2 + ((u_3 - y_2)m_3 \bmod^{\pm} p_3)\, p_1 p_2 \\ \vdots \quad \vdots \\ u = y_s = y_{s-1} + ((u_s - y_{s-1})m_s \bmod^{\pm} p_s)\, p_1 \cdots p_{s-1} \end{cases}$$

- Range analysis
  - Compute worst-case intervals from input intervals
  - Precise operations and roots of unity

Results

Saber Results: `MatrixVectorMul` and `InnerProd`

Table 1: Cycles for `MatrixVectorMul` and `InnerProd` in Saber.

`MatrixVectorMul`					
	Cortex-M4			Skylake (AVX2)	
	[BMKV20]	**Our Work**		[BMKV20]	**Our Work**	
$l = 2$	159k	66k	(− 58%)	7 002	5 215	(−25%)
$l = 3$	317k	125k	(− 61%)	14 145	9 579	(−32%)
$l = 4$	528k	205k	(− 61%)	24 342	14 959	(−39%)
`InnerProduct`					
	Cortex-M4			Skylake (AVX2)	
	[BMKV20]	**Our Work**		[BMKV20]	**Our Work**	
$l = 2$	73k	41k	(− 44%)	4 016	2 125	(−47%)
$l = 3$	99k	57k	(− 42%)	5 977	2 706	(−55%)
$l = 4$	126k	73k	(− 42%)	8 040	3 278	(−60%)

Saber Results: Full Scheme

Table 2: Clock cycles for `Lightsaber`, `Saber`, and `Firesaber`.

		Cortex-M4		Skylake (AVX2)	
		[BMKV20]	**Our Work**	[BMKV20]	**Our Work**
`Lightsaber`	**K**	466k	360k (−23%)	61 325	59 831 (−2%)
	E	653k	513k (−21%)	75 876	72 473 (−4%)
	D	678k	498k (−27%)	70 228	64 859 (−8%)
`Saber`	**K**	853k	658 (−23%)	104 832	99 71 (−5%)
	E	1 103k	864 (−22%)	125 835	118 44 (−6%)
	D	1 127k	835 (−26%)	118 553	107 26 (−10%)
`Firesaber`	**K**	1 340k	1 008k (−25%)	157 915	148 729 (−6%)
	E	1 642k	1 255k (−24%)	184 322	171 993 (−7%)
	D	1 679k	1 227k (−27%)	177 864	159 950 (−10%)

NTRU Results: Polynomial Multiplications

Table 3: Clock cycles for big by small polynomial multiplication in NTRU.

	Cortex-M4			Skylake (AVX2)		
n	[KRS19]	**Our Work**		[ZCH+19]	**Our Work**	
509	104k	101k	(− 3%)	6 643	8 540	(+29%)
677	175k	156k	(− 11%)	11 103	10 373	(−7%)
701	173k	156k	(− 10%)	11 242	10 373	(−8%)
821	230k	199k	(− 13%)	15 507	13 247	(−15%)

NTRU Results: Full Scheme

Table 4: Clock cycles for NTRU.

		Cortex-M4		Skylake (AVX2)	
		[KRS19]	**Our Work**	[ZCH+19]	**Our Work**
ntruhps2048509	**K**	79 682k	79 660k (±0%)	208 653	218 887 (+5%)
	E	572k	564k (−1%)	71 018	73 176 (+3%)
	D	545k	538k (−1%)	38 950	42 953 (+10%)
ntruhps2048677	**K**	143 808k	143 725k (±0%)	332 906	333 278 (±0%)
	E	849k	821k (−3%)	96 293	95 953 (±0%)
	D	845k	818k (−3%)	59 169	58 406 (−1%)
ntruhrss701	**K**	154 477k	154 403k (±0%)	299 066	298 505 (±0%)
	E	403k	377k (−6%)	56 616	56 084 (−1%)
	D	896k	871k (−3%)	62 503	61 199 (−2%)
ntruhps4096821	**K**	208 953k	207 495k (−1%)	458 614	451 664 (−2%)
	E	1 069k	1 027k (−4%)	114 986	113 935 (−1%)
	D	1 075k	1 030k (−4%)	74 182	70 917 (−4%)

LAC Results: Polynomial Multiplications

- $q = 251$ for `LAC-{128, 192, 256}-v3a`
- $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$
- Parameter n varies for security levels:
  - `LAC-128-v3a`: $n = 512$
  - `LAC-192-v3a` and `LAC-256-v3a`: $n = 1024$
- One of the multiplicands is ternary, i.e. coefficients are in $\{\pm 1, 0\}$

Table 5: LAC polynomial multiplication clock cycles on Cortex-M4 and Skylake

	Cortex-M4			Skylake (AVX2)		
	[LLZ+18]	**Our Work**		[LLZ+18]	**Our Work**	
`LAC-128-v3a`	638k	65k	(−90%)	14 691	4 552	(−69%)
`LAC-192-v3a`	1 274k	131k	(−90%)	73 955	10 119	(−86%)
`LAC-256-v3a`	1 701k	132k	(−92%)	73 955	10 119	(−86%)

LAC Results: Full Scheme

Table 6: Performance results in clock cycles for LAC

		Cortex-M4		Skylake (AVX2)	
		[LLZ+18]	**Our Work**	[LLZ+18]	**Our Work**
LAC-128-v3a	**K**	850k	282k (−67%)	53 000	42 167 (−20%)
	E	1 430k	450k (−69%)	76 418	59 252 (−22%)
	D	1 960k	565k (−71%)	86 209	55 880 (−35%)
LAC-192-v3a	**K**	1 507k	373k (−75%)	96 270	41 713 (−57%)
	E	2 427k	610k (−75%)	128 342	67 732 (−47%)
	D	3 329k	824k (−75%)	189 660	74 393 (−61%)
LAC-256-v3a	**K**	2 020k	459k (−77%)	143 568	76 917 (−46%)
	E	3 633k	748k (−79%)	202 346	106 836 (−47%)
	D	5 327k	1 111k (−79%)	262 901	104 897 (−60%)

Conclusion

- Saber
  - Coefficient ring: NTT-unfriendly
  - Polynomial modulus: NTT-friendly
  - `MatrixVectorMul`: NTT-friendly
- NTRU
  - Coefficient rings: NTT-unfriendly
  - The degrees of polynomials: large enough for NTTs
- LAC
  - Coefficient rings: NTT-unfriendly
  - Polynomial modulus: NTT-friendly
  - The degrees of polynomials: large enough for NTTs
- Compute the result as if $\mathbb{Z}$ is the coefficient ring
  - Cortex-M4: 32-bit prime
  - AVX2: two 16-bit primes

Works Worth Noting

- We optimized Saber on Cortex-M4 only for speed. We thank Michiel van Beirendonck for integrating stack optimizations. See commit `992f0f226503d43b6d33278ecb60a9168ed8d787` in `pqm4`.
- For even more stack optimized Saber on Cortex-M4 and more analysis on NTTs:
  - "Multi-moduli NTTs for Saber on Cortex-M3 and Cortex-M4"
  - `https://eprint.iacr.org/2021/995`
- For NTTs with 64-bit Armv8-A:
  - "Neon NTT: Faster Dilithium, Kyber, and Saber on Cortex-A72 and Apple M1"
  - `https://eprint.iacr.org/2021/986`
  - Barrett multiplication: 3-instruction single-width modular multiplication
  - Asymmetric multiplication: multiplying in $R[x]/\langle x^{\{2,4\}} - \zeta \rangle$ essentially as in $R[x]/\langle x^{\{2,4\}} - 1 \rangle$ *without* requiring $\zeta = \omega^{\{2,4\}}$, applicable whenever incomplete NTTs are re-used, e.g. Kyber and Saber

Thank you for your attention

Reference i

Jose Maria Bermudo Mera, Angshuman Karmakar, and Ingrid Verbauwhede.
Time-memory trade-off in toom-cook multiplication: an application to module-lattice based cryptography.
IACR Transactions on Cryptographic Hardware and Embedded Systems, 2020(2):222–244, Mar. 2020.

Matthias J. Kannwischer, Joost Rijneveld, and Peter Schwabe.
Faster multiplication in $\mathbf{Z}_{2^m}[x]$ on cortex-m4 to speed up NIST PQC candidates.
In *Applied Cryptography and Network Security*, pages 281–301, 2019.

Xianhui Lu, Yamin Liu, Zhenfei Zhang, Dingding Jia, Haiyang Xue, Jingnan He, and Bao Li.
LAC: practical ring-lwe based public-key encryption with byte-level modulus.
IACR Cryptol. ePrint Arch., 2018.
https://eprint.iacr.org/2018/1009.

Zhenfei Zhang, Cong Chen, Jeffrey Hoffstein, William Whyte, John M. Schanck, Andreas Hulsing, Joost Rijneveld, Peter Schwabe, and Oussama Danba.
NTRU.
Technical report, National Institute of Standards and Technology, 2019.
available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.