Short-Limb Modular Multiplication Techniques: Barrett, Montgomery, Plantard, and the Explicit CRT

Bo-Yin Yang
Institute of Information Science, Academia Sinica, Taipei, Taiwan
byyang@iis.sinica.edu.tw

August 20, 2026

Abstract

This note collects, in compressed form, some techniques for modular multiplication with word-size ("short-limb"), or at most a-handful-of-words sized moduli as they are used in implementations of lattice-based cryptography: Barrett reduction and multiplication (in signed and unsigned flavors, with exact error, range, and canonicality analyses), Montgomery reduction and multiplication (including the folded-constant form, the precise equivalence with Barrett multiplication, even moduli, the multi-limb case, and the k-reduction), Plantard multiplication (the original unsigned algorithm, the signed variant, and a variant taking signed inputs to the canonical unsigned representative in $[0, q)$), and modular multiplication via the explicit Chinese remainder theorem. These are compressed out of my lecture slides in the class Post-Quantum Cryptography at National Taiwan University 2020–2025 (EE 5176/921 U2540). All numerical examples, ranges, and windows stated here have been verified by exhaustive or randomized machine search; several constants and ranges correct typos and miscalculations that circulated after lectures.

1 Notation

Throughout, ℓ denotes a limb (word) size, typically $\ell \in \{16, 32\}$, and $R = 2^k$ a power of two, often $k = \ell$ or $k = 2\ell$. The modulus is M (or q when it is a small prime as in lattice-based schemes) and is odd unless stated otherwise.

For $m > 0$ we write $x \bmod^+ m$ for the representative of x in $[0, m)$ and $x \bmod^\pm m$ for the representative in $[-m/2, m/2)$. We write $\lfloor x \rfloor$, $\lceil x \rceil$ for floor and ceiling, and $\lfloor x \rceil := \left\lfloor x + \frac{1}{2} \right\rfloor$ for rounding half up; note $x \bmod^\pm m = x - m \lfloor x/m \rceil$. More generally, an *integer approximation* is any map $[\![\cdot]\!] : \mathbb{Q} \to \mathbb{Z}$ with $|x - [\![x]\!]| \le 1$ for all x; the three maps just named are integer approximations, and we write $x \bmod^{[\![]\!]} m := x - m\,[\![x/m]\!]$ for the associated signed remainder.

For a power of two 2^ℓ we abbreviate

$$[A]^\ell := \left\lfloor A/2^\ell \right\rfloor, \qquad [A]_\ell := A \bmod^\pm 2^\ell, \qquad [A]_\ell^+ := A \bmod^+ 2^\ell.$$

On a two's-complement machine $[A]^\ell$ is the arithmetic right shift, $[A]_\ell$ is truncation to ℓ bits read as a signed integer, and $[A]_\ell^+$ is the same bit pattern read unsigned. We write $\mathrm{Mulhi}_k(A, B) = [AB]^k = \left\lfloor AB/2^k \right\rfloor$ and $\mathrm{Mullo}_k(A, B) = [AB]_k = AB \bmod^\pm 2^k$ for the high and low halves of a product; most architectures provide both in some form.

2 Barrett reduction

2.1 Signed and unsigned variants

Let $\overline{M} := \left\lfloor 2^k/M \right\rceil$. The (signed, rounding) *Barrett reduction* [2] of A is

$$\mathrm{BAR}_M(A) \;:=\; A - M\left\lfloor \frac{A\overline{M}}{2^k} \right\rceil \;\approx\; A - M\left\lfloor \frac{A}{M} \right\rceil \;=\; A \bmod^{\pm} M.$$

On a machine, $\left\lfloor A\overline{M}/2^k \right\rceil = (A\overline{M} + 2^{k-1}) \gg k$ with an arithmetic shift.

If instead the unsigned $A \bmod M = A - \lfloor A/M \rfloor M$ is wanted, two natural choices of constant present themselves:

- With $\left\lfloor 2^k/M \right\rfloor$ and a truncating estimate, $A - M\left\lfloor A\left\lfloor 2^k/M \right\rfloor/2^k \right\rfloor$ is nonnegative for $A \geq 0$, but is not canonical: substituting $A = M$ returns M rather than 0.
- With $\widetilde{M} := \left\lceil 2^k/M \right\rceil$ the estimate can overshoot and the result can be negative. The output first becomes negative at

$$A_M \;=\; \left\lceil a \cdot 2^k/\widetilde{M} \right\rceil, \qquad a \;=\; \left\lceil \left(M - 2^k/\widetilde{M}\right)^{-1} \right\rceil,$$

  the first point where $A\widetilde{M}/2^k$ climbs just past an integer a while $A < aM$ still holds.

When out of bounds, one adjusts by $\pm M$ (a conditional addition or subtraction).

Example 2.1. Let $M = 4591$, $k = 32$. Then $\left\lfloor 2^k/M \right\rceil = \left\lceil 2^k/M \right\rceil = 935519$ and $\left\lfloor 2^k/M \right\rfloor = 935518$. Signed: $\mathrm{BAR}_M(2295) = 2295$ but $\mathrm{BAR}_M(2296) = -2295$. Unsigned with $\widetilde{M} = 935519$: inputs 4591, 4590, -4591 give 0, 4590, 4591; with the floor constant 935518: inputs 4591, 4592 give 4591, 1. For $\widetilde{M}$ the first negative output occurs at $a = 2161$, $A_M = 9\,921\,150 < M^2$, where the output is -1; every $0 \leq A < A_M$ gives a nonnegative output (verified by sweep).

2.2 Error decomposition

Write $\varepsilon_0 = M\overline{M}/2^k - 1$ (the relative error of the precomputed constant), and for a given A let ε_1 be the rounding error of the estimate and ε_2 that of the ideal quotient. Then

$$A \bmod^{\pm} M - \mathrm{BAR}_M(A) \;=\; M(\varepsilon_1 + \varepsilon_2) + A\varepsilon_0,$$

i.e. two bounded "random" terms plus a *drift* term growing linearly in A. The drift term is what eventually pushes BAR_M off the canonical value and, later, out of any fixed range.

2.3 Range

$\mathrm{BAR}_M(A)$ is piecewise linear in A with slope 1, dropping by M exactly at the points $A_j = \left\lceil (j2^k - 2^{k-1})/\overline{M} \right\rceil$ where the rounded estimate increments. Hence on any range $|A| \leq A_{\max}$ its extreme values are attained at the points $A_j - 1$, A_j, or at $\pm A_{\max}$, and can be enumerated exactly.

Example 2.2. For $M = 4591$: with $k = 32$ and $|A| \leq 2^{31}$ the range is ± 2512; with $k = 32$ and $|A| \leq 2^{32}$ it is ± 2728 (attained at $A = -4\,294\,965\,001$); with $k = 15$ and $|A| \leq 2^{15}$ it is ± 2881.

2.4 Canonical range

Proposition 2.3. *If* $|A| < 1/(2|\varepsilon_0|)$ *then* $\mathrm{BAR}_M(A) = A \bmod^{\pm} M$.

Proof. $A\overline{M}/2^k = (A/M)(1+\varepsilon_0)$, so the argument of the rounding differs from A/M by $|A||\varepsilon_0|/M < 1/(2M)$. Since M is odd, $2A - (2j+1)M$ is odd for every j, so A/M is at distance at least $1/(2M)$ from every half-integer, with the distance-$1/(2M)$ case approached only from one side; hence both arguments round to the same integer. □

Example 2.4. $M = 4591$. For $k = 32$: $\varepsilon_0 = 433/2^{32} \approx +1.01 \times 10^{-7}$, so BAR_M is canonical for $|A| < 4.96 \times 10^6$; the first non-canonical input is $|A| = 4\,960\,575$. For $k = 31$: $\varepsilon_0 = -2079/2^{31} \approx -9.7 \times 10^{-7}$, canonical range $|A| < 5.16 \times 10^5$, first failure at $|A| = 516\,488$. (The $k = 31$ constant corrects an exponent typo, -0.97×10^{-7}, in earlier lecture material; the resulting canonical range is $\approx 5 \times 10^5$, not 5×10^6.)

2.5 Instruction-set remarks

How the estimate is computed depends heavily on the architecture. ARM's `SMMULR` computes $(AB + 2^{31}) \gg 32$, giving centered 32-bit Barrett directly with $\overline{M} = \lfloor 2^{32}/M \rceil$. `SMLAWx` (x $\in \{\mathrm{B}, \mathrm{T}\}$) computes $\lfloor A\,B_{\mathrm{x}}/2^{16} \rfloor + C$, so a rounding estimate is obtained by seeding the accumulator with 2^{15}. On x86, `VPMULHRSW` computes $(XY + 2^{14}) \gg 15$ on 16-bit lanes. Many architectures provide a plain multiply-return-high Mulhi_k; precomputing $M' = \lfloor 2^{k+e}/M \rceil$ with $2^e < M < 2^{e+1}$ one uses $\lfloor A/M \rfloor \approx \mathrm{Mulhi}_k(A, M') \gg e$.

3 Montgomery reduction and multiplication

3.1 Seiler (Signed Montgomery) Reduction

Barrett reduction needs a full-length product with the precomputed constant. Montgomery's alternative [10] computes not $A \bmod M$ but $A/R \bmod M$, where $R = 2^k$ is coprime to the odd M. Precompute $M' = M^{-1} \bmod^{\pm} R$; given A, set $t = (A \bmod^{\pm} R)\, M' \bmod^{\pm} R$, so that $tM \equiv A \pmod R$; then $A - tM$ is divisible by R and

$$\mathrm{MR}(A) \;:=\; \frac{A - tM}{R} \;\equiv\; A\,R^{-1} \pmod M.$$

Only the top half of tM is needed.

Example 3.1. $M = 83$, $R = 100$, $M' = 47$. For $A = 6412$: $t = 12 \cdot 47 \bmod^{\pm} 100 = -36$, and $(6412 + 36 \cdot 83)/100 = 9400/100 = 94$. Note $94 > M$: Montgomery reduction does not return the canonical representative. For $A = 3322$: $t = 22 \cdot 47 \bmod^{\pm} 100 = 34$ and $(3322 - 34 \cdot 83)/100 = 5$.

M' is computed by Hensel lifting $x \mapsto x(2 - Mx)$, which doubles the number of correct bits each step.

Example 3.2. $M = 47$, $R = 256$: starting from $x_0 = 1$ (correct mod 2), reducing centered at each step gives $x_1 = -1 \pmod 4$, $x_2 = -1 \pmod{16}$, $x_3 = -49 \pmod{256}$, and indeed $47 \cdot (-49) \equiv 1 \pmod{256}$.

3.2 Unsigned Montgomery reduction

For unsigned data take $M' = -M^{-1} \bmod^{+} R$ and $t = (A \bmod R)M' \bmod R$, so that $A + tM \equiv 0 \pmod R$ and $\mathrm{MR}(A) = (A + tM)/R$. This handles unsigned (hence multiprecision) operands at the cost of larger intermediate values and a full-length addition.

Example 3.3. $M = 83$, $R = 100$, $M' = 53$: $A = 6412$ gives $(6412 + 36 \cdot 83)/100 = 94$ as before, while $A = 3322$ gives $(3322 + 66 \cdot 83)/100 = 88$, congruent but not equal to the signed result 5.

3.3 Range

In the signed version $|\mathrm{MR}(A)| = |A - tM|/R \le |A|/R + M/2$; in particular $|A| < RM/2$ implies $|\mathrm{MR}(A)| < M$. In the unsigned version $0 \le \mathrm{MR}(A) = (A + tM)/R < A/R + M \le 2M$ for $0 \le A < RM$. Both bounds shrink with $|A|$.

3.4 Multiplication by a constant, folded

If b is known one computes $ab \pmod M$ in the Montgomery domain as $\mathrm{MR}(a \cdot B)$ with $B = bR \bmod^{\pm} M$ precomputed. On architectures where high and low halves of products are separate instructions this folds further:

Proposition 3.4. *Let $B = bR \bmod^{\pm} M$ and $B' = BM' \bmod^{\pm} R$ with $M' = M^{-1} \bmod^{\pm} R$. Then*

$$\mathrm{MR}(aB) \;=\; \mathrm{Mulhi}_k(a, B) \;-\; \mathrm{Mulhi}_k\big(M,\ \mathrm{Mullo}_k(a, B')\big),$$

where Mullo_k returns the centered residue $\bmod^{\pm} R$. For $|a| \le R/2$ and $|b| \le M/2$ the result lies in $(-M, M)$ and is congruent to $ab\,R^{-1} \cdot R \equiv ab \pmod M$.

Proof. $t = (aB)M' \bmod^{\pm} R = a(BM') \bmod^{\pm} R = aB' \bmod^{\pm} R$. Then $\mathrm{MR}(aB) = (aB - tM)/R$, and since $aB \equiv tM \pmod R$ the two floors $\lfloor aB/R \rfloor$ and $\lfloor tM/R \rfloor$ discard equal low parts, so their difference equals the exact quotient. The range is the signed range bound above. (Verified over 3×10^4 random instances at $\ell = 16$.) □

This is Seiler's two-high-multiplications [12] form used in current fast Kyber and Dilithium NTT code: with B, B' precomputed, a constant multiplication costs one Mullo and two Mulhi. In particular the AArch64 and x86_64 NTT kernels of `mlkem-native` [9] build on this form, and are proved functionally correct (as well as memory-safe and constant-time) at the object-code level in HOL Light [6], using the `s2n-bignum` verification infrastructure.

3.5 The Montgomery domain

Writing $X^{(M)} := XR \bmod M$, one has $\mathrm{MR}\big(a^{(M)} b^{(M)}\big) \equiv abR = c^{(M)}$ and $a^{(M)} + b^{(M)} \equiv c^{(M)}$ for $c = a + b$: all arithmetic may be carried out on Montgomery representatives. Conversion into the domain is $a^{(M)} = \mathrm{MR}\big(a \cdot (R^2 \bmod M)\big)$ with $R^2 \bmod M$ precomputed.

3.6 Montgomery multiplication is Barrett multiplication

The two families coincide exactly; the idea was first due to Nick (Shang-Yi) Yang and first proved by myself; the equivalence has since been machine-checked, in Isabelle/HOL [3]. For any integer approximation $[\![\cdot]\!]$ define the *Barrett multiplication*

$$\mathrm{BAR}^{[\![]\!]}_M(A, B) \;:=\; AB \;-\; M \left\lfloor \frac{A\,[\![BR/M]\!]}{R} \right\rfloor.$$

Lemma 3.5. $[\![R/M]\!] \bmod^{\pm} R = \big(-(R \bmod^{[\![]\!]} M)\,(M^{-1} \bmod^{\pm} R)\big) \bmod^{\pm} R$, *and likewise with* R *replaced by* BR.

Proof. $M\,[\![BR/M]\!] = BR - (BR \bmod^{[\![]\!]} M)$; reduce $\bmod^{\pm} R$ and multiply by $M^{-1} \bmod^{\pm} R$. □

Theorem 3.6. *With* MR *the signed Montgomery reduction,*

$$\mathrm{BAR}^{[\![]\!]}_M(A,B) \;=\; \mathrm{MR}\big(A\,(BR \bmod^{[\![]\!]} M)\big),$$

except in the boundary case $A\,[\![BR/M]\!] \equiv R/2 \pmod R$, *where the two sides differ by* M.

Proof. Let $c = [\![BR/M]\!]$ and $X = A(BR - Mc) = A\,(BR \bmod^{[\![]\!]} M)$. Then

$$\mathrm{BAR}^{[\![]\!]}_M(A,B) = AB - \frac{M}{R}\Big(Ac - (Ac \bmod^{\pm} R)\Big) = \frac{1}{R}\Big(X + M\,(Ac \bmod^{\pm} R)\Big),$$

while $\mathrm{MR}(X) = \big(X - (XM' \bmod^{\pm} R)M\big)/R$ with $XM' \equiv A(BR - Mc)M' \equiv -Ac \pmod R$. Since $(-Ac) \bmod^{\pm} R = -(Ac \bmod^{\pm} R)$ unless $Ac \equiv R/2 \pmod R$ (where centered negation wraps), the two expressions agree outside that case.[1] □

Thus "Montgomery vs. Barrett" for word-size moduli is a choice of representation of the same map, and mixed pipelines are consistent.

3.7 Montgomery with Even Moduli

Let $M = 2^t u$ with u odd, $R = 2^k > 2^t$, and $u' = u^{-1} \bmod R$. Precompute $B = bR \bmod M$ and $B' = u' \cdot \frac{B - bR}{2^t} \bmod R$ (note $2^t \mid B - bR$ since $M \mid B - bR$). Then with $t_a = aB' \bmod R$ one has $R \mid aB - t_a M$ and $(aB - t_a M)/R \equiv ab \pmod M$: the quotient is correct modulo u by the usual Montgomery argument, and modulo 2^t because $B - B'M \equiv bR \pmod{2^t R}$ by construction of B'. (Verified over 2×10^4 random instances.)

3.8 Extending to Modern Multi-limb Montgomery Reduction

Let $R = B^r$ for a small base B (e.g. $B = 2^w$) and $M' = -M^{-1} \bmod B$. Iterate $A_0 = A$; $t_i = (A_i \bmod B)M' \bmod B$; $A_{i+1} = (A_i + t_i M)/B$ for $0 \le i < r$. Then $A_r \equiv A/R \pmod M$ and $0 \le A_r < 2M \forall r$ if $0 \le A < MR$, since $A_r = (A + tM)/R$ with $t = t_0 + t_1 B + \cdots + t_{r-1} B^{r-1} < R$.

Example 3.7. Figure 1 traces $A = 765\,846$, $M = 997$, $R = 1000$, $B = 10$, $M' = 7$: the three steps clear one trailing digit each, with t-digits $2, 8, 2$ and intermediate values $76\,784$ and $8\,476$; indeed $765\,846/1000 \equiv 1047 \equiv 50 \pmod{997}$.

Interleaving these reduction steps with the schoolbook multiplication that produced A gives the coarsely/finely integrated operand scanning (CIOS/FIOS) family; e.g. multiplying $942 \times 813 = 765\,846$ with reduction steps merged into the partial-product accumulation yields the same 1047 while saving temporary storage (coarse integration) or storage and time (fine integration, merging the two inner loops).

[1] The reader is encouraged to run randomized checks; I ran one in class giving 7 failures in 120,000 trials, every one at the excluded boundary.

0	7	6	5	8	4	6	$i=0:\ t_0 \equiv 6\times 7 \equiv 2 \pmod{10}$
			1	9	9	4	$+\,2\times 997$
0	7	6	7	8	4		$i=1:\ t_1 \equiv 4\times 7 \equiv 8$
		7	9	7	6		$+\,8\times 997$
0	8	4	7	6			$i=2:\ t_2 \equiv 6\times 7 \equiv 2$
	1	9	9	4			$+\,2\times 997$
1	0	4	7				

Figure 1: Multi-limb Montgomery reduction of $A = 765\,846$ with $M = 997$, $R = 1000$, $B = 10$, $M' = -1/M \equiv 7 \pmod{10}$. Each step adds the multiple $t_i M$ of the modulus that clears the current trailing digit; dropping that digit is the division by B, visible as the right-to-left staircase. After $r = 3$ steps the value $1047 \equiv A/R \pmod{997}$ remains.

3.9 The k-reduction for $M = \kappa \cdot 2^m + 1$

For moduli of the form $M = \kappa \cdot 2^m + 1$ with small odd κ (e.g. $12289 = 3 \cdot 2^{12} + 1$), take $R = 2^m$; then $M \equiv 1 \pmod{R}$, so the unsigned Montgomery multiplier is $M' = -1$ and $t = -A \bmod R$. Writing $A = A_H R + A_L$,

$$-\kappa A \;\equiv\; A/R \;\equiv\; \mathrm{MR}(A) \;=\; A_H - \kappa A_L \pmod{M},$$

using $\kappa R \equiv -1$, i.e. $R^{-1} \equiv -\kappa$. Applying it twice to $A = A_H R^2 + A_M R + A_L$ gives $\kappa^2 A \equiv A_H - \kappa A_M + \kappa^2 A_L \pmod{M}$. No multiplication by a precomputed constant is needed at all.

4 Barrett Multiplication

Barrett multiplication by a constant B (the signed version of Shoup multiplication [13], in the integer-approximation formulation of [4]) precomputes $c = [\![BR/M]\!]$ and evaluates $\mathrm{BAR}_M^{[\![]\!]}(A,B) = AB - M\,\lfloor Ac/R \rceil$; on a machine the products AB and the estimate live in single words, and with a multiply-subtract the cost is three multiplications for a variable B, two of them independent.

Proposition 4.1. *For any integer approximation,*

$$\left|\mathrm{BAR}_M^{[\![]\!]}(A,B)\right| \;\le\; \frac{\left|A\,(BR \bmod^{[\![]\!]} M)\right|}{R} + \frac{M}{2}.$$

Proof. $AB - M\,\lfloor Ac/R \rceil = A(BR - Mc)/R + M\big(Ac/R - \lfloor Ac/R \rceil\big)$, the last being in $[-1/2, 1/2)$. □

Theorem 4.2. *Let $c = \lfloor BR/M \rceil$, $\varepsilon = |c - BR/M|$, and let h be maximal with $\varepsilon \le 2^{-h}$ (if $\varepsilon = 0$ the conclusion below holds for all A). If $|A| < 2^{k+h-1}/M$ then $\mathrm{BAR}_M^{\lfloor\rceil}(A,B) = AB \bmod^{\pm} M$.*

Proof. Ac/R differs from AB/M by $|A|\varepsilon/R < 2^{k+h-1} \cdot 2^{-h}/(MR) = 1/(2M)$; as in Proposition 2.3, AB/M is at distance at least $1/(2M)$ from every half-integer, so the two arguments round alike. (In-class python check: 0 violations in $16,000$ boundary-biased trials.) □

Our canonical range *grows* with the accuracy h of the precomputed constant; $k = (\ell - 1) + \lfloor \log_2 M \rfloor - \lceil \log_2 |B| \rceil$ keeps c and the products within ℓ-bit words while maximizing $k + h$.

5 Plantard multiplication

Plantard [11] observed that for word-size moduli one of the three multiplications above can be saved when one operand is a constant: precomputing bq' with $q' = q^{-1} \bmod 2^{2\ell}$ turns the whole reduction into one $\ell \times 2\ell$-bit product followed by one short multiplication with an additive correction. We give the original unsigned algorithm, the signed variant, and a variant with signed input and canonical unsigned output. Throughout, q is odd and q' denotes an inverse of q modulo $2^{2\ell}$; all three algorithms output a representative of $ab\,(-2^{-2\ell}) \bmod q$, the "Plantard domain" analog of the Montgomery factor.

5.1 Unsigned Plantard multiplication

Theorem 5.1 (essentially [11])**.** *Let $\varphi = (1+\sqrt{5})/2$ and let $q < 2^\ell/\varphi$ be odd, $q' = q^{-1} \bmod^+ 2^{2\ell}$. For $a, b \in [0, q]$ put $p = abq' \bmod^+ 2^{2\ell}$ and*

$$r \;=\; \Big[\big([\,p\,]^\ell + 1\big)\,q\Big]^\ell \;=\; \left\lfloor \frac{\big(\lfloor p/2^\ell \rfloor + 1\big)\,q}{2^\ell} \right\rfloor .$$

Then $r = ab\,(-2^{-2\ell}) \bmod^+ q$, and moreover $r \in [0, q-1]$.

Proof. Write $p = p_1 2^\ell + p_0$ with $p_1 = \lfloor p/2^\ell \rfloor \in [0, 2^\ell)$, $p_0 \in [0, 2^\ell)$. Since $pq \equiv ab \pmod{2^{2\ell}}$, the integer $I = (pq - ab)/2^{2\ell}$ satisfies $I \equiv ab(-2^{-2\ell}) \pmod q$, and

$$r = \left\lfloor \frac{pq - p_0 q + q2^\ell}{2^{2\ell}} \right\rfloor = I + \left\lfloor \frac{ab - p_0 q + q2^\ell}{2^{2\ell}} \right\rfloor .$$

The bracket vanishes: on the one hand $ab - p_0 q + q2^\ell \geq q2^\ell - (2^\ell - 1)q = q > 0$; on the other, $ab + q2^\ell \leq q^2 + q2^\ell < 2^{2\ell}$, the last inequality being exactly $q < 2^\ell(\sqrt{5}-1)/2 = 2^\ell/\varphi$ (the positive root of $q^2 + q2^\ell - 2^{2\ell}$) — this is where the golden ratio enters. So $r = I$; finally $-1 < -q^2/2^{2\ell} \leq I < q\,2^{2\ell}/2^{2\ell} = q$, giving $r \in [0, q-1]$. □

Remark 5.2. [11] states the output range as $[0, q]$; the endpoint q is producible by the displayed formula (at $p_1 = 2^\ell - 1$) but, as the proof shows, is never reached from inputs in $[0, q]$: the output is already the canonical representative in $[0, q-1]$, and no final conditional subtraction is needed. (Verified exhaustively at $\ell = 8$ for several admissible q.)

Remark 5.3. The bound $q < 2^\ell/\varphi$ is sufficient but not sharp for fixed small q: at $\ell = 8$ the threshold is $2^8/\varphi \approx 158.2$, yet $q = 159$ passes an exhaustive test over all $a, b \in [0, q]$, and the first failing odd modulus is $q = 161$, at $(a, b) = (152, 160)$.

5.2 Signed Plantard multiplication

Signed arithmetic removes an addition from each NTT butterfly, so a signed variant is what implementations want [12, 1, 7]. The following formulation, with a general redundancy parameter $\bar{q}$, also explains where the additive constant comes from.

Theorem 5.4. *Let q be odd and $\bar{q} \geq 2$ an integer with $q\bar{q} < 2^{\ell-1}$; let $q' = q^{-1} \bmod^\pm 2^{2\ell}$. For signed $a, b \in [-q\bar{q},\, q\bar{q}]$ put $p = abq' \bmod^\pm 2^{2\ell}$ and*

$$r \;=\; \Big[\big([\,p\,]^\ell + \bar{q}\big)\,q\Big]^\ell .$$

Then $r = ab\,(-2^{-2\ell}) \bmod^{\pm} q$*, and moreover* $r \in \left[-\frac{q-1}{2}, \frac{q-1}{2}\right]$*: the output is the canonical centered representative. Indeed, we could do (cf. [8])* $r = \left\lfloor [\,p\,]^{\ell}\, q/2^{\ell} \right\rceil$ *instead.*

Proof. Write $p = p_1 2^{\ell} + p_0$, now with $p_1 = [\,p\,]^{\ell} \in [-2^{\ell-1}, 2^{\ell-1})$ and $p_0 \in [0, 2^{\ell})$ (the arithmetic shift always leaves a nonnegative low part). As before $I = (pq - ab)/2^{2\ell}$ is an integer congruent to $ab(-2^{-2\ell})$, and

$$r = I + \left\lfloor \frac{ab - p_0 q + q\bar{q}\, 2^{\ell}}{2^{2\ell}} \right\rceil .$$

The bracket vanishes: $ab - p_0 q + q\bar{q}2^{\ell} \geq q\bar{q}2^{\ell} - q^2\bar{q}^2 - (2^{\ell} - 1)q = q\big((\bar{q} - 1)2^{\ell} - q\bar{q}^2 + 1\big) > 0$, since $q\bar{q}^2 < \bar{q}\, 2^{\ell-1} \leq (\bar{q} - 1)2^{\ell}$ for $\bar{q} \geq 2$; and $ab + q\bar{q}2^{\ell} \leq q^2\bar{q}^2 + q\bar{q}2^{\ell} < 2^{2\ell-2} + 2^{2\ell-1} = \frac{3}{4}\, 2^{2\ell}$. Hence $r = I = \dfrac{p_1 q}{2^{\ell}} - \dfrac{ab - p_0 q}{2^{2\ell}}$, where the first term lies in $[-q/2,\, q/2 - q/2^{\ell}]$ and the second in $(-1/2,\, 1/4)$: indeed $ab - p_0 q < ab \leq q^2\bar{q}^2 < 2^{2\ell-2}$, while $ab - p_0 q > -(q^2\bar{q}^2 + q2^{\ell}) \geq -(q\bar{q}\, 2^{\ell-1} + q\bar{q}\, 2^{\ell-1}) = -q\bar{q}\, 2^{\ell} > -2^{2\ell-1}$, using $q\bar{q} < 2^{\ell-1}$ for the first and third comparisons and $\bar{q} \geq 2$ for the middle one. Therefore $-\frac{q}{2} - \frac{1}{4} < r < \frac{q}{2} + \frac{1}{2}$, and as r is an integer and q odd, $r \in [-(q-1)/2,\, (q-1)/2]$. □

Remark 5.5. Taking $\bar{q} = 2^{\alpha}$ recovers Algorithm 10 of [7] with its constraint $q < 2^{\ell-\alpha-1}$. That paper states the output range as $[-\frac{q+1}{2}, \frac{q}{2})$, which is what the crude bound $p_1 \geq -2^{\ell-1}$ yields; the left endpoint $-\frac{q+1}{2}$ requires $|ab| \geq 2^{2\ell-1}$ and is therefore unreachable under the input constraint $|ab| \leq (q\bar{q})^2 < 2^{2\ell-2}$ — the reachable range is the tight, symmetric one above. (Verified exhaustively at $\ell = 8$ for $(q, \bar{q}) \in \{(31, 4), (41, 3), (11, 11), (63, 2)\}$ and at $\ell = 16$ for $q = 3329$, $\bar{q} = 8$, where the range is exactly ± 1664.) The tight range matters for lazy reduction: coefficients grow by $q/2$ per Cooley–Tukey layer instead of q.

Remark 5.6. On the Cortex-M4 this multiplication by a constant costs two instructions — `SMULWB` for the 16×32-bit product $[abq']_{2\ell}]^{\ell}$ and `SMLABB` to fold the multiplication by q, the additive constant $q\bar{q}$, and the final shift (read from the top half) [7]. This is the concrete payoff of saving one multiplication: Montgomery and Barrett constant multiplications need three. On SIMD architectures without an $\ell \times 2\ell$-bit multiply the product must be emulated from high/low halves, and the advantage narrows or disappears.

5.3 Signed input, unsigned output

Implementations that keep NTT coefficients signed still need nonnegative canonical values at the boundary — for serialization, compression, or an unsigned base multiplication — and [7] note that a separate Barrett pass remains necessary there. The following variant removes that pass: it accepts signed input and returns the canonical representative in $[0, q)$ directly, at the same cost as the signed algorithm. The idea is to read the high part of p *unsigned* and let a final unsigned high multiplication produce the range $[0, q)$ structurally.

Theorem 5.7. *Let* q *be odd,* $0 < q < 2^{\ell}$*,* $q' = q^{-1} \bmod^{+} 2^{2\ell}$*, and let* $s \geq 1$ *be an integer. For* $c \in \mathbb{Z}$ *put*

$$p = cq' \bmod^{+} 2^{2\ell}, \qquad x = \left(\left\lfloor p/2^{\ell} \right\rceil + s\right) \bmod^{+} 2^{\ell}, \qquad Z = \left\lfloor \frac{x\, q}{2^{\ell}} \right\rceil .$$

Then:

(i) $Z \in [0, q-1]$ *for* every $c \in \mathbb{Z}$*, unconditionally;*

(ii) if $(2^\ell-1)q - sq2^\ell \le c < 2^{2\ell} - sq2^\ell$*, then* $Z \equiv c\,(-2^{-2\ell}) \pmod q$*, and hence* $Z = c\,(-2^{-2\ell}) \bmod^+ q$ *is the canonical unsigned representative.*

Proof. (i) is immediate: $x \in [0, 2^\ell)$, so $0 \le Z \le \lfloor (2^\ell-1)q/2^\ell \rfloor = q-1$.

For (ii), write $p = m2^\ell + p_0$ with $m = \lfloor p/2^\ell \rfloor$, $p_0 \in [0, 2^\ell)$, and note $pq \equiv c \pmod{2^{2\ell}}$ with $p \ge 0$, so $I = (pq-c)/2^{2\ell}$ is an integer congruent to $c(-2^{-2\ell})$. If $m + s < 2^\ell$ (no wrap), then

$$Z = \left\lfloor \frac{(m+s)q}{2^\ell} \right\rfloor = \left\lfloor \frac{pq - p_0 q + sq2^\ell}{2^{2\ell}} \right\rfloor = I + \left\lfloor \frac{c - p_0 q + sq2^\ell}{2^{2\ell}} \right\rfloor,$$

and the hypothesis of (ii) makes the bracket vanish for every $p_0 \in [0, 2^\ell)$. If $m + s \ge 2^\ell$ the wrap replaces x by $x - 2^\ell$, lowering Z by exactly q; the congruence is unaffected and (i) still pins Z into $[0, q-1]$. □

Corollary 5.8 (multiplication by a constant)**.** *Let* $w \in [0, q)$ *and precompute* $W = \big((-2^{2\ell}w) \bmod^+ q\big) \cdot q' \bmod^+ 2^{2\ell}$*. For signed* a *with* $|a| \le 2^{\ell-1}$*, running Theorem 5.7 on* $p = aW \bmod^+ 2^{2\ell}$ *returns* $Z = aw \bmod^+ q$*, provided* $2 \le s$ *and* $q(s+1) \le 2^\ell$*. In particular* $s = \lfloor 2^{\ell-1}/q \rfloor$ *works for all* $q < 2^{\ell-1} \cdot \frac{2}{3}$*, and centers the validity window.*

Proof. The folded constant contributes the factor $-2^{2\ell}$ that cancels the Plantard domain, so $Z \equiv aw$. The window of Theorem 5.7 must cover $|c| \le q2^{\ell-1}$: the left endpoint requires $sq2^\ell \ge (2^\ell-1)q + q2^{\ell-1}$, i.e. $s \ge 2$, and the right endpoint $q2^{\ell-1} < 2^{2\ell} - sq2^\ell$, i.e. $q(s+\frac{1}{2}) < 2^\ell$. □

At $\ell = 16$ this has been verified exhaustively — every 16-bit signed a against every constant $w \in [0, q)$ — for $q = 3329$ ($s = 10$), $q = 7681$ ($s = 4$), and $q = 12289$ ($s = 3$): zero failures, output exactly $[0, q-1]$ in each case. Note that $q = 12289$ is fully covered here even though its products $|c| \le q2^{15}$ overflow the validity windows of some other 16-bit Plantard variants.

Remark 5.9 (AVX2 realization)*.* Take $\ell = 16$; every `ymm` register holds sixteen independent 16-bit lanes, and each line below acts per lane. The symbols are as follows. Per modulus: q odd and s an integer with $2 \le s$ and $q(s+1) \le 2^{16}$ (canonically $s = \lfloor 2^{15}/q \rfloor$: $3329 \mapsto 10$, $7681 \mapsto 4$, $12289 \mapsto 3$); $q' = q^{-1} \bmod^+ 2^{32}$; `Q` and `S` are vectors with q, resp. s, in every lane. Per constant multiplicand $w \in [0, q)$: the folded constant $b = (-2^{32}w) \bmod^+ q$ of Corollary 5.8, the 32-bit word $W = bq' \bmod^+ 2^{32}$, and its signed split

$$L = W \bmod^\pm 2^{16}, \qquad H = \big(\lfloor W/2^{16} \rfloor + [L < 0]\big) \bmod^\pm 2^{16},$$

so that $W \equiv H \cdot 2^{16} + L \pmod{2^{32}}$; the carry $[L < 0]$ re-absorbs the sign extension of L, and since H is only ever used modulo 2^{16} its own possible wrap is harmless. The input a is a signed 16-bit lane, any value in $[-2^{15}, 2^{15})$. Write $p = aW \bmod^+ 2^{32}$ (never materialized) and $p_0 = p \bmod^+ 2^{16}$ for its discarded low half, which appears only in the analysis.

`vpmulhw` t, a, L	$t = \lfloor aL/2^{16} \rfloor$, the exact signed high half
`vpmullw` u, a, H	$u = aH \bmod 2^{16}$, low half (bit pattern, sign-blind)
`vpaddw` m, t, u	$m = (t+u) \bmod 2^{16} = \lfloor p/2^{16} \rfloor$ read *unsigned*
`vpaddw` x, m, S	$x = (m+s) \bmod 2^{16}$; a wrap lowers Z by exactly q
`vpmulhuw` Z, x, Q	$Z = \lfloor xq/2^{16} \rfloor$, both operands unsigned

The third line is valid because $aW \equiv aH \cdot 2^{16} + aL \pmod{2^{32}}$, so bits [31:16] of p equal $\big(aH + \lfloor aL/2^{16} \rfloor\big) \bmod 2^{16}$. By Theorem 5.7 and Corollary 5.8, $Z \in [0, q-1]$ unconditionally and

$Z = aw \bmod^+ q$ for every 16-bit signed a. The final `VPMULHUW` is what makes the construction work twice over: its unsigned product-high is structurally confined to $[0, q-1]$, and its shift count 16 (rather than the 15 of `VPMULHRSW`) restores the natural $16+16 = 2\ell$ shift budget, so no extra shift instruction is needed. Under the opposite inverse convention $q' = -q^{-1} \bmod^+ 2^{32}$, only the fold changes, to $b = (+2^{32}w) \bmod^+ q$. For comparison, producing the same signed-to-$[0, q)$ map from a Montgomery or Barrett multiplication costs a trailing conditional correction (e.g. `vpsubw`/`vpminuw`) on top of the multiplication itself.

6 Multiplication via the explicit CRT

If $p_1, \ldots, p_s$ are pairwise coprime, the Chinese remainder theorem reconstructs x from the residues $x \bmod p_i$ uniquely modulo $P = \prod p_i$. For multiplication one computes modulo several word-size primes and reconstructs; the reconstructions below are "explicit" in that they avoid multiprecision division.

6.1 Divided-difference form

Theorem 6.1. *Let $p_i > 0$ be odd and pairwise coprime, and let $|u_i| < p_i/2$ be given residues. With $m_i = (p_1 \cdots p_{i-1})^{-1} \bmod p_i$ define*

$$y_1 = u_1, \qquad y_i = y_{i-1} + \big((u_i - y_{i-1})\, m_i \bmod p_i\big)\, p_1 \cdots p_{i-1} \quad (2 \le i \le s),$$

all inner reductions centered. Then $u = y_s$ satisfies $u \equiv u_i \pmod{p_i}$ for all i and $|u| < P/2$.

Proof. By induction: $y_i \equiv y_{i-1} \pmod{p_j}$ for $j < i$, while modulo p_i the correction contributes $(u_i - y_{i-1}) m_i \prod_{j<i} p_j \equiv u_i - y_{i-1}$. For the size, if $|y_{i-1}| < \frac{1}{2}\prod_{j<i} p_j$ then $|y_i| \le \frac{1}{2}\big(\prod_{j<i} p_j - 1\big) + \frac{1}{2}(p_i - 1)\prod_{j<i} p_j = \frac{1}{2}\big(\prod_{j\le i} p_j - 1\big)$. □

Example 6.2. $p = (3, 5, 7)$, $u = (2, 3, 2)$: $m_2 = 2$, $m_3 = 1$, and $y_2 = 2 + ((3-2)\cdot 2 \bmod 5)\cdot 3 = 8$, $u = y_3 = 8 + ((2-8)\cdot 1 \bmod 7)\cdot 15 = 23$.

Example 6.3. With the NTT-friendly primes $p_1 = 7681$, $p_2 = 10753$, $p_3 = 11777$ $(= 23\cdot 2^9 + 1)$: $m_2 = -5373$, $m_3 = -703$. For $u = (220,\ -2124,\ 4411)$:

$$y_2 = 220 + \big((-2344)(-5373) \bmod^{\pm} 10753\big)\cdot 7681 = 19\,579\,089,$$

$$u = y_3 = y_2 + \big((4411 - y_2)(-703) \bmod^{\pm} 11777\big)\cdot 7681 \cdot 10753 = -156\,413\,064\,853.$$

The reconstruction can be carried out entirely in single-word arithmetic by folding the Montgomery factor into the precomputed constants. With $R = 2^{16}$ and $\mathrm{MR}(x, \pi) = \big(x - ((x \bmod^{\pm} R)(\pi^{-1} \bmod^{\pm} R) \bmod^{\pm} R)\pi\big)/R$, precompute $m_2 R \bmod p_2 = 3563$, $p_1 R \bmod p_3 = -2295$, $p_1 R \bmod p = 1821$, $m_3 R \bmod p_3 = -184$, and $p_1 p_2 R \bmod p = 598$, where $p = 4591$ is the target modulus. Then, continuing Example 6.3 and reducing u modulo p without ever leaving 16-bit products:

$$\begin{aligned}
(u_2 - y_1)m_2 &\equiv \mathrm{MR}\big((-2344)\cdot 3563,\, p_2\big) = 2549 \pmod{p_2},\\
y_2 &\equiv 220 + \mathrm{MR}\big(2549\cdot(-2295),\, p_3\big) = 5715 \pmod{p_3},\\
y_2 &\equiv 220 + \mathrm{MR}\big(2549\cdot 1821,\, p\big) = -1526 \pmod{p},\\
(u_3 - y_2)m_3 &\equiv \mathrm{MR}\big((4411 - 5715)\cdot(-184),\, p_3\big) = -1894 \pmod{p_3},\\
u &\equiv -1526 + \mathrm{MR}\big((-1894)\cdot 598,\, p\big) = 465 \pmod{p},
\end{aligned}$$

which agrees with $-156\,413\,064\,853 \bmod^{\pm} 4591 = 465$.

6.2 The explicit CRT

Theorem 6.4 ([5])**.** *Let $p_1, \dots, p_s$ be pairwise coprime and positive, $P = \prod p_i$; let q_i satisfy $q_i P/p_i \equiv 1 \pmod{p_i}$; let $|u| < P/2$ with residues $u_i \equiv u \pmod{p_i}$; and let t_i be any integers with $t_i \equiv u_i q_i \pmod{p_i}$. Then, with $\alpha = \sum_i t_i/p_i$,*

$$u \;=\; P\alpha - P\,\lfloor \alpha \rceil\,.$$

Proof. $P\alpha = \sum_i t_i(P/p_i) \equiv u_j q_j (P/p_j) \equiv u \pmod{p_j}$ for every j, so $P\alpha \equiv u \pmod{P}$; thus $r = \alpha - u/P \in \mathbb{Z}$, and $|\alpha - r| = |u/P| < 1/2$ forces $r = \lfloor \alpha \rceil$. □

The rounding of α can be computed from low-precision fixed-point approximations:

Theorem 6.5. *Let $\beta_1, \dots, \beta_s$ be reals and r an integer with $|r - \sum_i \beta_i| < \frac{1}{4}$, and let $2^a \geq 2s$. Then $r = \left\lfloor \frac{3}{4} + 2^{-a} \sum_i \lfloor 2^a \beta_i \rfloor \right\rfloor$.*

Proof. $r < \frac{1}{4} + \sum \beta_i \leq \frac{1}{4} + 2^{-a}(s + \sum \lfloor 2^a \beta_i \rfloor) \leq \frac{3}{4} + 2^{-a} \sum \lfloor 2^a \beta_i \rfloor \leq \frac{3}{4} + \sum \beta_i < r + 1$. □

If additionally $|u| < P/4$ in Theorem 6.4, then $|\,\lfloor \alpha \rceil - \sum t_i/p_i| < 1/4$ and Theorem 6.5 computes $\lfloor \alpha \rceil$ from the truncations $\lfloor 2^a t_i/p_i \rfloor$.

Example 6.6. $p = (3, 5, 7)$, $u = (2, 3, 2)$: $q = (-1, 1, 1)$, and one may take $t = (-2, -2, 2)$. Then $\alpha = -\frac{2}{3} - \frac{2}{5} + \frac{2}{7} = -\frac{82}{105}$, $\lfloor \alpha \rceil = -1$, and $u = 105\alpha + 105 = 23$. With $a = 3$, the truncations $\lfloor 2^3 t_i/p_i \rfloor$ are $-6, -4, 2$, summing to -8, and $\left\lfloor \frac{3}{4} - \frac{8}{8} \right\rfloor = -1 = \lfloor \alpha \rceil$.

6.3 Explicit CRT modulo M

Theorem 6.7. *In the setting of Theorem 6.4, with the t_i chosen centered ($t_i = u_i q_i \bmod^{\pm} p_i$), let $M > 0$ and*

$$v \;=\; \sum_i t_i \left(P/p_i \bmod^{\pm} M\right) \;-\; \left(P \bmod^{\pm} M\right) \lfloor \alpha \rceil\,.$$

Then $v \equiv u \pmod{M}$ and $|v| \leq \left(\frac{s+1}{2} + \frac{1}{4} \sum_i p_i\right) M$.

Proof. Modulo M, $v \equiv \sum t_i(P/p_i) - P\,\lfloor \alpha \rceil = u$. For the size: $|t_i| \leq p_i/2$ and each centered residue is at most $M/2$ in magnitude, so the sum is at most $\frac{M}{4} \sum p_i$; and $|\,\lfloor \alpha \rceil\,| \leq \frac{s+1}{2}$ since $|\alpha| \leq s/2$, giving the second term at most $\frac{(s+1)M}{4}$. (The stated, slightly looser bound follows; verified with 6000 random instances.) □

Corollary 6.8. *$v \equiv \sum_i t_i \,(P/p_i \bmod M \bmod p_j) - (P \bmod M \bmod p_j)\, \lfloor \alpha \rceil \pmod{p_j}$: the reduction of u modulo M can itself be computed residue-wise.*

Example 6.9. $p = (3, 5, 7)$, $u = (2, 3, 2)$, $t = (-2, -2, 2)$, $\lfloor \alpha \rceil = -1$, $M = 4$: $v = (-2)(35 \bmod^{\pm} 4) + (-2)(21 \bmod^{\pm} 4) + 2(15 \bmod^{\pm} 4) + (105 \bmod^{\pm} 4) = -1 \equiv 23 \pmod{4}$. The output need not be centered: for $u \equiv (1, 2, 3)$ — i.e. $u = 52$ — one gets $t = (-1, 2, 3)$, $\lfloor \alpha \rceil = 0$, and with $M = 8$: $v = (-1)(3) + 2(-3) + 3(-1) = -12 \equiv 52 \pmod{8}$, with $|-12| > M/2$.

Acknowledgments

This note grew out of lecture material presented at NTUEE. Numerical claims were re-verified by machine search in the course of preparing this version; discrepancies with my lecture slides floating around in the internet (noted inline where they occur, as in Example 2.4) are resolved in favor of the verified values. I thank my students for bearing with me through extreme tedium and even learning something. Claude Opus 5 corrected my typos and grammatical errors; most of its stylistic suggestions I turned down (so the stilted English is my own fault). I had Fable 5 go over my derivations, but all errors are still my responsibility.

References

[1] Daichi Aoki, Kazuhiko Minematsu, Toshihiko Okamura, Tsuyoshi Takagi. Efficient word size modular multiplication over signed integers. *ARITH 2022* (29th IEEE Symposium on Computer Arithmetic), 94–101.

[2] Paul Barrett. Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor. In *Advances in Cryptology — CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 311–323. Springer, 1987.

[3] Hanno Becker. Neon NTT – (auto)formalised. IACR Cryptology ePrint Archive, Report 2026/1223. Isabelle/HOL sources in the `mldsa-native` repository, `https://github.com/pq-code-package/mldsa-native`, under `proofs/`.

[4] Hanno Becker, Vincent Hwang, Matthias J. Kannwischer, Bo-Yin Yang, and Shang-Yi Yang. Neon NTT: Faster Dilithium, Kyber, and Saber on Cortex-A72 and Apple M1. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(1):221–244, 2022.

[5] Daniel J. Bernstein and Jonathan P. Sorenson. Modular exponentiation via the explicit Chinese remainder theorem. *Mathematics of Computation*, 76(257):443–454, 2007.

[6] John Harrison. HOL Light: an overview. In *Theorem Proving in Higher Order Logics (TPHOLs 2009)*, volume 5674 of *Lecture Notes in Computer Science*, pages 60–66. Springer, 2009. `https://github.com/jrh13/hol-light`.

[7] Junhao Huang, Jipeng Zhang, Haosong Zhao, Zhe Liu, Ray C. C. Cheung, Çetin Kaya Koç, and Donglong Chen. Improved Plantard arithmetic for lattice-based cryptography. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(4):614–644, 2022. Also IACR Cryptology ePrint Archive, Report 2022/956.

[8] Vincent Hwang. Software Implementations of Polynomial Multiplications for Lattice-Based Cryptosystems, Ph.D. Thesis, Radboud University, 2026, `doi.org/10.54195/9789465152516`

[9] The `mlkem-native` contributors. `mlkem-native`: secure, fast, and portable C90 implementation of ML-KEM (FIPS 203). `https://github.com/pq-code-package/mlkem-native`; formal proofs under `proofs/hol_light`, built on the `s2n-bignum` infrastructure, `https://github.com/awslabs/s2n-bignum`.

[10] Peter L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44(170):519–521, 1985.

[11] Thomas Plantard. Efficient word size modular arithmetic. *IEEE Transactions on Emerging Topics in Computing*, 9(3):1506–1518, 2021.

[12] Gregor Seiler. Faster AVX2 optimized NTT multiplication for Ring-LWE lattice cryptography. IACR Cryptology ePrint Archive, Report 2018/039, 2018.

[13] Victor Shoup. NTL: A library for doing number theory. `https://libntl.org`.