PQC Final Project

NTT multiplication in NTRUHPS2048677

B07901100 林亮昕
D09921011 洪逸霖

1. Specification

   PQC Algorithm: NTRUHPS2048677

   Equipment: Cortex-m4

2. Our NTT multiplication

   2.1 Prelimanaries

   $p = 12902401 = 1440 * 256 * 35 + 1$

   $N = 1440$

   $\zeta = 2577$ (where $\zeta^{1440} = 1$ in Z_p)

   2.2 Twisted version

$$\frac{Z[x]}{x^{677}-1} \to \frac{Z_p[x]}{x^{1440}-1} \xrightarrow{8-NTT} \bigcup_{i=1}^{8} \frac{Z_p[x]}{x^{180}-c_i^6} \xrightarrow{twisted} \bigcup_{i=1}^{8} \frac{Z_p[y]}{y^{180}-1}$$

$$\xrightarrow{6-NTT} \bigcup_{j=1}^{8}\bigcup_{i=1}^{6} \frac{Z_p[y]}{y^{30}-d_i^6} \xrightarrow{twisted} \bigcup_{j=1}^{8}\bigcup_{i=1}^{6} \frac{Z_p[z]}{z^{30}-1} \xrightarrow{6-NTT} \bigcup_{k=1}^{8}\bigcup_{j=1}^{6}\bigcup_{i=1}^{6} \frac{Z_p[z]}{z^{5}-e_i}$$

   2.3 None-twisted version

$$\frac{Z[x]}{x^{677}-1} \to \frac{Z_p[x]}{x^{1440}-1} \xrightarrow{8-NTT} \bigcup_{i=1}^{8} \frac{Z_p[x]}{x^{180}-c_i^6} \xrightarrow{6-NTT} \bigcup_{i=1}^{48} \frac{Z_p[x]}{x^{30}-d_i^6} \xrightarrow{6-NTT} \bigcup_{i=1}^{288} \frac{Z_p[x]}{x^{5}-e_i}$$

   2.4 Radix-8-NTT

   We apply three radix-2 NTT to realize readix-8 NTT. Cooley-Tukey butterfly structure are used in the forward direction and Gentleman-Sande butterfly structure are used in backward direction.

   2.5 Radix-6-NTT

   For twisted version, we apply Good's Trick to radix-6 NTT:

$$\frac{Z_P[x]}{x^6-1} \cong \frac{Z_P[x]}{x-1} \times \frac{Z_P[x]}{x+1} \times \frac{Z_P[x]}{x-\omega} \times \frac{Z_P[x]}{x+\omega} \times \frac{Z_P[x]}{x-\omega^2} \times \frac{Z_P[x]}{x+\omega^2}$$

($\omega \neq 1$ is the third of 1 in Z_P)

For None-twisted version, we apply Good's Trick with special "rotation":

$$\frac{Z_P[x]}{x^6 - c^6} \cong \frac{Z_P[x]}{x - cy, y^6 - 1}$$

$$\cong \frac{Z'_P[y]}{y-1} \times \frac{Z'_P[y]}{y+1} \times \frac{Z'_P[y]}{y-\omega} \times \frac{Z'_P[y]}{y+\omega} \times \frac{Z'_P[y]}{y-\omega^2} \times \frac{Z'_P[y]}{y+\omega^2}$$

Coefficients in $Z'_P[y]$ are those in $Z_p[x]$ "rotated" by some power of c:

$$\sum_{i=0}^{5} a_i x^i \in Z_P[x] \rightarrow \sum_{i=0}^{5} a_i (cy)^i = \sum_{i=0}^{5} (a_i c^i) y^i \in Z'_P[y]$$

2.6 Multiplication in $\frac{Z_p[x]}{x^5 - e_i}$

We apply school book multiplication in $\frac{Z_p[x]}{x^5 - e_i}$.

3. Code optimization

3.1 Radix-8 NTT in forward direction

Since multiplicands in ntruhpw2048677 has degree at most 676, the first radix-2 NTT layer is redundant. Now consider $x^{720} - 1$ and $x^{720} + 1$. We have

$$x^{720} - 1 = (x^{360} - 1)(x^{360} + 1)$$

$$= (x^{180} - 1)(x^{180} + 1)(x^{180} - \zeta^{360})(x^{180} + \zeta^{360})$$

$$x^{720} + 1 = (x^{360} - \zeta^{360})(x^{360} + \zeta^{360})$$

$$= (x^{180} - \zeta^{180})(x^{180} + \zeta^{180})(x^{180} - \zeta^{540})(x^{180} + \zeta^{540})$$

($\zeta = 2577, \zeta^{1440} = 1 \; in \; Z_p$)

Note that coefficients in ntruhps2048677 are between -1024 and 1024 and $|1024 * \zeta^{360}| = |1024 * (-57584)| < 2^{32}$. Therefore we can perform direct multiplication (mul.w) instead of montgomery multiplication in Coleey-Tukey butterfly for this case. For ζ^{180} and ζ^{540}, we store them in floating point registers for montgomery reduction.

3.2 Radix-6 NTT:

For Good's trick, we only need to store ω and ω^2 in registers. Moreover, with the following property:

$$1 + \omega + \omega^2 = 0$$

, we are able to calculate

$$(a_0 + a_3) + (a_4 + a_1)\omega^2 + (a_2 + a_5)\omega$$
$$= 3[(a_0 + a_3) + (a_4 + a_1) + (a_2 + a_5)] - (a_0 + a_3) + (a_4 + a_1)\omega + (a_2 + a_5)\omega^2$$

in our Good's trick. This helps us reduce code size and increase the speed. Moreover, the technique can also be used in none-twisted version since coefficient in none-twisted version are just the coefficient in twisted version rotated by some constant.

4. Correctness:

Since $p = 12902401 > 1024 * 677 * 2 + 1 = 1386497$, the overall scheme is correct. In the radix-8 NTT, the maximum possible value is $|1024 * -57584 * 4| < 2^{32}$ and the maximum times of addition without overflow in this layer is $\frac{2^{32}}{1024*57584} \approx 72.8$, which is much larger than the number of addition in our code.

In the second and third layer (radix-6 NTT), the maximum possible value is $12902401 * 288 < 2^{32}$ and the maximum times of addition without overflow is $\frac{2^{32}}{12902401} \approx 332.8$, which is also larger than the number of addition in our code.

Therefore, we can always derive the correct answer in out NTT multiplication scheme.

5. Performance:

5.1 NTT multiplication in ntruhps2048677

Multiplication version	Baseline	[KRS19]	[CHK+21]	Our work Twisted	Our work None-twisted	Our work None- twisted
NTT trick	No	No	Yes	Yes	Yes	Yes
NTT domain	No	No	1536	1440	1440	1440
Trick	SchoolBook	2*Toom-4	3*3-layer-radix-2	8-NTT +2*Good's trick +2*twisted	8-NTT +2*Good's trick	8-NTT +2*Good's trick
Base Mul	677*677	11*11	3*3	5*5	5*5	5*5
Loop unrolling				Yes	Yes	Only Base Mul and Last

						packing stage
Code size					2.6MB	987KB
Cycle count	4591k	175k	156k	153k	149k	176k

5.2 Ntruhps2048677

(Note: the gcc-arm-none-ebai version is 10.2.1)

(Note: the pqm4 version is about one month ago since the latest version cannot run properly on our macOS Sierra 10.13.6 version)

NTRU2048677 version	pqclean	m4f w/ our NTT multiplication	m4f	m4f w/ our NTT multiplication (loop unrolling)
Correctness	OK	OK	OK	OK
KeyGen	122985923	143796812	143740651	143719310(-0.015%)
Enc	1802981	846571	827937	820735(-0.87%)
Dec	4287767	833488	814810	807652(-0.88%)

6. Conclusion

Out NTT multiplication scheme with loop unrolling is slightly faster than other multiplication scheme. However, the code size grows large since the number of loop is significant. Our optimization in 6-radix NTT shows that Good's trick can be applied to $\frac{Z_P[x]}{x^6-c^6}$ with a special rotation technique. Furthermore, the calculation in tradition Good's trick can be speed up using the property $1+\omega+\omega^2=0$. On the other hand, 8-radix NTT can also be speed up as some Montgomery multiplications can be substituted by normal multiplication. Finally, our ntruhps2048677 speeds up about 0.015% in KeyGen, 0.87% in Enc, and 0.88% in Dec compared to m4f implementation.

7. Reference

[KRS19] Faster multiplication in Z2m[x] on Cortex-M4 to speed up NIST PQC candidates

[CHK+21]NTT Multiplication for NTT-unfriendly Rings New Speed Records for Saber and NTRU on Cortex-M4 and AVX2