Front. Comput. Sci., 2024, 18(4): 184818
https://doi.org/10.1007/s11704-024-3288-9

LETTER

Generalized splitting-ring number theoretic transform

Zhichuang LIANG[1], **Yunlei ZHAO** (✉)[1,2], **Zhenfeng ZHANG**[3]

1 School of Computer Science, Fudan University, Shanghai 200433, China
2 State Key Laboratory of Cryptology, Beijing 100036, China
3 Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

1 Introduction

Lattice-based cryptography is one of the most promising routine of post-quantum cryptography (PQC). The fundamental and time-consuming operation in lattice-based schemes is the polynomial multiplication in cyclotomic ring $\mathbb{Z}_q[x]/(\Phi_m(x))$. Most schemes utilize power-of-two cyclotomic rings, where $m = 2^k, k \geqslant 1$, $n = \varphi(m) = m/2$, $\varphi(\cdot)$ is the Euler function and $\Phi_m(x) = x^n + 1$, and trinomial cyclotomic rings, where $m = 2^k 3^l, k, l \geqslant 1$, $n = \varphi(m) = m/3$ and $\Phi_m(x) = x^n - x^{n/2} + 1$.

Number theoretic transform (NTT) is a special case of fast Fourier transforms (FFT) over a finite field [1,2]. FFT/NTT are the most efficient methods for computing polynomial multiplication of high degree, due to their quasilinear complexity $O(n \log n)$. In recent years, some literatures have explored methods for weakening parameter restrictions and further improving the overall performance of the aforementioned NTT algorithms. As for power-of-two cyclotomics, Zhu et al. [3] proposed the Karatsuba-NTT (K-NTT), which relaxes the restrictions on the modulus q from $q \equiv 1 \ (\mathrm{mod}\ m)$ to $q \equiv 1 \ (\mathrm{mod}\ \frac{m}{2^\alpha})$. Liang et al. [4] presented further optimization to $q \equiv 1 \ (\mathrm{mod}\ \frac{m}{2^{\alpha+\beta}})$ and proposed Hybrid-NTT (H-NTT). Additionally, they extended these techniques to trinomial cyclotomic rings $\mathbb{Z}_q[x]/(\Phi_{3^2 2^k}(x))$, and proposed another NTT variant named G3-NTT. We found that K-NTT, H-NTT, and G3-NTT can be considered as special cases of a more general algorithm that encompasses all such "splitting-ring-based" NTT algorithms.

Our contributions in this paper are listed as follows.

(1) We propose the first **G**eneralized **S**plitting-**R**ing **N**umber **T**heoretic **T**ransform, referred to as GSR-NTT, and demonstrate that K-NTT, H-NTT, and G3-NTT can be regarded as special cases of GSR-NTT under different parameterizations.

(2) We introduce a succinct methodology for complexity analysis, based on which our GSR-NTT can derive its optimal parameter settings.

(3) We apply our GSR-NTT to accelerate polynomial multiplications in the lattice-based scheme NTTRU [5] and power-of-three cyclotomic rings.

2 Concrete construction

2.1 Splitting-ring isomorphism

Let $m = \prod_{i=1}^{k} p_i^{e_i}$ be the unique factorization of m where $\{p_i\}_{i=0}^{k}$ are some prime numbers and $e_i \geqslant 1$ for all i. Let $z = \prod_{i=1}^{k} p_i^{e_i'}$ where $1 \leqslant e_i' \leqslant e_i$ for all i. We describe our splitting-ring isomorphism Ψ as follows, whose basic idea is similar to parts of Nussbaumer's trick [6], but our splitting-ring isomorphism is more general and can be applied to more types of underlying polynomial rings.

$$\Psi : \mathbb{Z}_q[x]/(\Phi_m(x)) \;\cong\; \Big(\mathbb{Z}_q[y]/(\Phi_z(y))\Big)[x]/(x^{\frac{m}{z}} - y),$$

$$f = \sum_{i=0}^{\varphi(m)-1} f_i x^i \;\mapsto\; \Psi(f) = \sum_{j=0}^{\frac{m}{z}-1} F_j x^j,$$

where $F_j = \sum_{i=0}^{\varphi(z)-1} f_{\frac{m}{z}\cdot i + j} y^i \in \mathbb{Z}_q[y]/(\Phi_z(y))$. Note that $\varphi(z)$-point (incomplete) NTT (whose forward/inverse transform are denoted by $(\mathcal{I})\mathcal{NTT}$) can be utilized as $\mathbb{Z}_q[y]/(\Phi_z(y)) \cong \prod_{k=0}^{\frac{\varphi(z)}{d}-1} \mathbb{Z}_q[y]/(y^d - \zeta^{\tau(k)})$, where $1 \leqslant d \leqslant \varphi(z)$, $d | \varphi(z)$, $d | z$; $q \equiv 1 \bmod \frac{z}{d}$; ζ is the primitive $\frac{z}{d}$-th root of unity in $\mathbb{Z}_q$; $\tau(k)$ is the power of ζ for the kth term (k starting from zero).

2.2 GSR-NTT descriptions

Our GSR-NTT to compute $h = f \cdot g \in \mathbb{Z}_q[x]/(\Phi_m(x))$ is described in Algorithms 1−3 as follows.

Algorithm 1 demonstrates the forward transform of GSR-NTT. Algorithm 2 presents the point-wise multiplication of GSR-NTT with the aid of one-iteration Karatsuba algorithm. Note that $\hat{y} := \mathcal{NTT}(y)$ is precomputed and stored. Finally, the inverse transform of GSR-NTT is shown in Algorithm 3.

2.3 Succinct methodology for complexity analysis

We present a concise methodology for analyzing the complexity of GSR-NTT. We consider the general number of forward transforms, point-wise multiplications, and inverse transforms in GSR-NTT, and denote them by l_F, l_M, and l_I, respectively. For the multiplication costs, we define the

Received April 12, 2023; accepted January 15, 2024

E-mail: ylzhao@fudan.edu.cn

Algorithm 1 The forward transform of GSR-NTT: ForTran

Input: $f \in \mathbb{Z}_q[x]/(\Phi_m(x))$ and underlying $\varphi(z)$-point $\mathcal{NTT}$
Output: $\hat{f} := \{\hat{F}_j\}_{j=0}^{\frac{m}{z}-1}$

Map f into $\Psi(f) = \sum_{j=0}^{\frac{m}{z}-1} F_j x^j$
for $j = 0, \ldots, \frac{m}{z} - 1$ **do**
  $\hat{F}_j := \mathcal{NTT}(F_j)$
end for
return $\{\hat{F}_j\}_{j=0}^{\frac{m}{z}-1}$

Algorithm 2 The Karatsuba-aid point-wise multiplication of GSR-NTT: PWM

Input: $\hat{f} := \{\hat{F}_j\}_{j=0}^{\frac{m}{z}-1}, \hat{g} := \{\hat{G}_j\}_{j=0}^{\frac{m}{z}-1}$ and underlying point-wise multiplication "$\circ$"
Output: $\hat{h} := \{\hat{H}_j\}_{j=0}^{\frac{m}{z}-1}$

$\hat{y} := \mathcal{NTT}(y)$
for $i = 0, \ldots, \frac{m}{z} - 1$ **do**
  $\hat{T}_i = \hat{F}_i \circ \hat{G}_i$
end for
for $i = 0, \ldots, \frac{m}{z} - 2$ **do**
  for $j = i, \ldots, \frac{m}{z} - 1$ **do**
    $\hat{R}_{i,j} := (\hat{F}_i + \hat{F}_j) \circ (\hat{G}_i + \hat{G}_j) - \hat{T}_i - \hat{T}_j$
  end for
end for
for $j = 0, \ldots, \frac{m}{z} - 1$ **do**
  $\hat{H}_j = \left(\sum_{\substack{l+k=j \\ 0\leqslant l<k\leqslant j}} \hat{R}_{l,k} + \sum_{\substack{l+k=j \\ 0\leqslant l=k\leqslant j}} \hat{T}_l \right) + \hat{y} \circ \left(\sum_{\substack{l+k=\frac{m}{z}+j \\ j+1\leqslant l<k\leqslant \frac{m}{z}-1}} \hat{R}_{l,k} + \sum_{\substack{l+k=\frac{m}{z}+j \\ j+1\leqslant l=k\leqslant \frac{m}{z}-1}} \hat{T}_l \right)$
end for
return $\{\hat{H}_j\}_{j=0}^{\frac{m}{z}-1}$

Algorithm 3 The inverse transform of GSR-NTT: InvTran

Input: $\hat{h} := \{\hat{H}_j\}_{j=0}^{\frac{m}{z}-1}$ and underlying $\varphi(z)$-point $\mathcal{INTT}$
Output: h

for $j = 0, \ldots, \frac{m}{z} - 1$ **do**
  $H_j := \mathcal{INTT}(\hat{H}_j)$
end for
$h := \Psi^{-1}(\sum_{j=0}^{\frac{m}{z}-1} H_j x^j)$
return h

corresponding costs of underlying $\mathcal{NTT}$ and $\mathcal{INTT}$ as $T_m(\mathcal{NTT})$ and $T_m(\mathcal{INTT})$ respectively. Therefore, the concrete multiplication complexity of GSR-NTT can be expressed as follows:

$$T_m(\text{GSR-NTT}) = l_F \cdot \frac{m}{z} \cdot T_m(\mathcal{NTT}) + l_I \cdot \frac{m}{z} \cdot T_m(\mathcal{INTT}) + l_M \cdot \left[\left(\frac{m}{z} + 1 \right) \cdot \frac{d+3}{4} + \left(1 - \frac{m}{z} - 2 \cdot \frac{z}{m} \right) \cdot \frac{1}{2d} \right] \cdot \varphi(m).$$

2.4 Instantiations

K-NTT [3] with $\alpha \in \{0, 1, \ldots, \log n - 1\}$, is a special case of GSR-NTT, by setting (m, q, z, d) of GSR-NTT to be $(2n, q, \frac{2n}{2^\alpha}, 1)$ where n is a power of two and $q \equiv 1 \bmod \frac{2n}{2^\alpha}$. Then, $\mathbb{Z}_q[x]/(x^n + 1) \cong \left(\mathbb{Z}_q[y]/(y^{\frac{n}{2^\alpha}} + 1) \right)[x]/(x^{2^\alpha} - y)$, and $\frac{n}{2^\alpha}$-point full NTT is described as: $\mathbb{Z}_q[y]/(y^{\frac{n}{2^\alpha}} + 1) \cong \prod_{k=0}^{\frac{n}{2^\alpha}-1} \mathbb{Z}_q[y]/(y - \zeta^{2\cdot \mathrm{br}_{n/2^\alpha}(k)+1})$, where ζ is the primitive $\frac{n}{2^{\alpha-1}}$-th root of unity in $\mathbb{Z}_q$.

H-NTT [4] with $\alpha \in \{0, 1, \ldots, \log n - 1\}, \beta \in \{0, 1, \ldots, \log \frac{n}{2^\alpha}\}$ can also be instantiated from GSR-NTT as follows. Let (m, q, z, d) of GSR-NTT be $(2n, q, \frac{2n}{2^\alpha}, 2^\beta)$ where $q \equiv 1 \bmod \frac{2n}{2^{\alpha+\beta}}$. Its splitting-ring isomorphism is the same as that of K-NTT. However, H-NTT applies $\frac{n}{2^\alpha}$-point incomplete NTT which is described as: $\mathbb{Z}_q[y]/(y^{\frac{n}{2^\alpha}} + 1) \cong \prod_{k=0}^{\frac{n}{2^{\alpha+\beta}}-1} \mathbb{Z}_q[y]/(y^{2^\beta} - \zeta^{2\cdot \mathrm{br}_{n/2^{\alpha+\beta}}(k)+1})$, where ζ is the primitive $\frac{n}{2^{\alpha+\beta-1}}$-th root of unity in $\mathbb{Z}_q$.

G3-NTT [4] with $\alpha \in \{0, 1, \ldots, \log \frac{n}{3} - 1\}, \beta \in \{0, 1, \ldots, \log \frac{n}{3\cdot 2^\alpha}\}$ operates over trinomial cyclotomic rings, which is derived by setting (m, q, z, d) of GSR-NTT to be $(3n, q, \frac{n}{2^\alpha}, 2^\beta)$. Then, $\mathbb{Z}_q[x]/(x^n - x^{n/2} + 1) \cong \left(\mathbb{Z}_q[y]/(y^{\frac{n}{3\cdot 2^\alpha}} - y^{\frac{n}{3\cdot 2^{\alpha+1}}} + 1) \right)[x]/(x^{3\cdot 2^\alpha} - y)$, and $\frac{n}{3\cdot 2^\alpha}$-point incomplete NTT is applied as: $\mathbb{Z}_q[y]/(y^{\frac{n}{3\cdot 2^\alpha}} - y^{\frac{n}{3\cdot 2^{\alpha+1}}} + 1) \cong \prod_{k=0}^{\frac{n}{3\cdot 2^{\alpha+\beta}}-1} \mathbb{Z}_q[y]/(y^{2^\beta} - \zeta^{\tau(k)})$, where ζ is the primitive $\frac{n}{2^{\alpha+\beta}}$-th root of unity in $\mathbb{Z}_q$.

Finally, we consider the application of GRT-NTT over power-of-three cyclotomic rings $\mathbb{Z}_q[x]/(\Phi_m(x))$, where $m = 3^k$, $k \geqslant 1$, $\Phi_m(x) = x^n + x^{n/2} + 1$, $n = 2m/3$. Let $\frac{m}{z} = 3^\alpha$, $d = 3^\beta$ and $q \equiv 1 \bmod \frac{m}{3^{\alpha+\beta}}$, where $\alpha \in \{0, 1, \ldots, \log_3 \frac{n}{2} - 1\}$ and $\beta \in \{0, 1, \ldots, \log_3 \frac{n}{2\cdot 3^\alpha}\}$. Then $\Phi_z(y) = y^{\frac{n}{3^\alpha}} + y^{\frac{n}{2\cdot 3^\alpha}} + 1$. Hence, $\mathbb{Z}_q[x]/(x^n + x^{n/2} + 1) \cong \left(\mathbb{Z}_q[y]/(y^{\frac{n}{3^\alpha}} + y^{\frac{n}{2\cdot 3^\alpha}} + 1) \right)[x]/(x^{3^\alpha} - y)$. $\frac{n}{3^\alpha}$-point NTT is applied as: $\mathbb{Z}_q[y]/(y^{\frac{n}{3^\alpha}} + y^{\frac{n}{2\cdot 3^\alpha}} + 1) \cong \prod_{k=0}^{\frac{n}{3^{\alpha+\beta}}-1} \mathbb{Z}_q[y]/(y^{3^\beta} - \zeta^{\tau(k)})$, where ζ is the primitive $\frac{n}{2\cdot 3^{\alpha+\beta-1}}$-th root of unity in $\mathbb{Z}_q$.

3 Applications and experiments

3.1 Application to NTTRU

We demonstrate how to utilize our GSR-NTT to accelerate the polynomial multiplication in NTTRU [5], an NTRU-based key encapsulation mechanism (KEM) over trinomial cyclotomic ring $\mathbb{Z}_{7681}[x]/\ (x^{768} - x^{384} + 1)$. Totally 6 forward transforms, 4 point-wise multiplications, and 1 inverse transform are required for the KEM scheme of NTTRU.

Applying the succinct methodology for complexity analysis of our GSR-NTT, we set $m = 2304$, $n = m/3$ and $q = 7681$, $l_F = 6$, $l_M = 4$ and $l_I = 1$. Then we search for the optimal (z, d) for the least computational complexity. To ensure better compatibility with the base case inversion algorithm in NTTRU, we final obtain $(z = 768, d = 1)$. In this case, the original NTT algorithm of NTTRU requires $\frac{3}{2} \log \frac{n}{3} + \frac{14}{3} n$ multiplications, while our GSR-NTT only requires $\frac{3}{2} \log \frac{n}{3} + \frac{11}{3} n$ multiplications, which is n multiplications less than that of NTTRU. As shown in Fig. 1, for the KEM scheme of NTTRU, our GSR-NTT achieves speed-ups of 24.7%, 37.6%, and 28.9% for the key generation, encapsulation, and decapsulation algorithms, respectively, leading to a total speed-up of 29.4%.

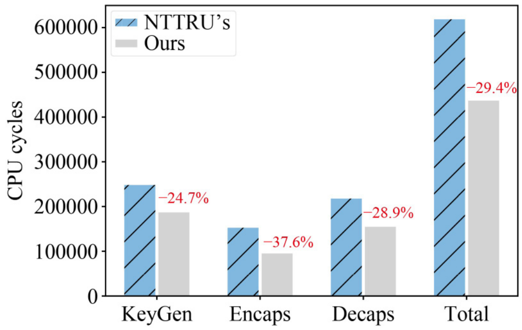

Fig. 1 Comparison between original NTT algorithm of NTTRU and our GSR-NTT for KEM schemes

3.2 Application to single polynomial multiplication

Here we primarily consider polynomial multiplications over power-of-three cyclotomic rings $\mathbb{Z}_q[x]/(\Phi_m(x))$ where $m = 3^k$, $k \geqslant 1$, $\Phi_m(x) = x^n + x^{n/2} + 1$, $n = 2m/3$ and $q \equiv 1 \bmod m$.

The prior NTT algorithm for computing polynomial multiplications in power-of-three cyclotomic rings is proposed in [7], which requires a total of $3n\log_3\frac{n}{2} + \frac{7}{2}n$ multiplications. As for our GSR-NTT, we search the parameters (z, d) for optimal multiplication complexity, and found that the multiplication complexity of GSR-NTT achieves its optimal value of $3n\log_3\frac{n}{2} + \frac{37}{18}n$ when using the setting $(m, q, z = \frac{m}{3}, d = 3)$, which has $\frac{13}{9}n$ less multiplications than that of [7].

4 Conclusions

In this paper, we propose GSR-NTT and demonstrate that K-NTT, H-NTT, and G3-NTT are specific instances of GSR-NTT. We introduce a succinct methodology for complexity analysis, and utilize our GSR-NTT to accelerate polynomial multiplications in NTTRU and power-of-three cyclotomic rings.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant No. 61877011), the National Key Research and Development Program of China (No. 2022YFB2701600), the Shanghai Science and Technology Innovation Action Plan (No. 21DZ2200500), and the Shandong Provincial Key Research and Development Program of China (Nos. 2017CXG0701 and 2018CXGC0701).

Competing interests The authors declare that they have no competing interests or financial conflicts to disclose.

References

1. Pollard J M. The fast Fourier transform in a finite field. Mathematics of Computation, 1971, 25(114): 365–374
2. Agarwal R C, Burrus C S. Number theoretic transforms to implement fast digital convolution. Proceedings of the IEEE, 1975, 63(4): 550–560
3. Zhu Y M, Liu Z, Pan Y B. When NTT meets Karatsuba: preprocess-then-NTT technique revisited. In: Proceedings of the 23rd International Conference on Information and Communications Security. 2021, 249−264
4. Liang Z C, Shen S Y, Shi Y T, Sun D N, Zhang C X, Zhang G Y, Zhao Y L, Zhao Z X. Number theoretic transform: generalization, optimization, concrete analysis and applications. In: Proceedings of the 16th International Conference on Information Security and Cryptology. 2020, 415−432
5. Lyubashevsky V, Seiler G. NTTRU: truly fast NTRU using NTT. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2019, 2019(3): 180–201
6. Nussbaumer H. Fast polynomial transform algorithms for digital convolution. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1980, 28(2): 205–215
7. Hassan C A, Yayla O. Radix-3 NTT-based polynomial multiplication for lattice-based cryptography. IACR Cryptology ePrint Archive, 2022, 726