

A-23 高速多項式乗算による格子暗号 CRYSTALS-Kyber の効率化に向けた検討

奥さくら*, 工藤桃成*

(*福岡工業大学情報工学部情報通信工学科)

1. はじめに

現在普及しているRSA暗号などは、量子計算機の発展により現実的な時間で解読可能になる危険性がある。そのため、量子計算機による解読にも耐えうる暗号(耐量子計算機暗号)が注目されている。中でもCRYSTALS-Kyber[1]は、多項式乗算に数論変換(NTT)を適用することで高速化が可能であるが、利用可能なパラメータに制約が生じる場合がある。先行研究[1]では、この制約を回避するため、Karatsuba法を用いた高速化が提案されている。一方で、Karatsuba法の一般化としてToom-Cook法が知られており、Karatsuba法と同様にパラメータの制約を回避した上でのさらなる高速化が理論上は期待できる。本研究では、このToom-Cook法をCRYSTALS-Kyberの多項式乗算に適用し、実験によってその有効性を検証する。

2. 格子暗号CRYSTALS-Kyber

n, k, m, q をパラメータとし(n :2冪の自然数, k, m :自然数, q :奇素数), $R_q = \mathbb{F}_q[x]/(x^n + 1)$ (剰余環)とおく。

■鍵生成 $A \in R_q^{m \times k}$ ($m \times k$ 行列)をランダムに生成する。 $s \in R_q^k$ (k 次元列ベクトル)と $e \in R_q^m$ (m 次元列ベクトル)を成分の各係数が十分小さい値を取るようサンプルする。 $t = A \cdot s + e \in R_q^m$ を計算し、組 (A, t) を公開鍵、ベクトル s を秘密鍵とする。

■暗号化 平文 $M \in R_q$ に対し, $r \in R_q^k$, $e_1 \in R_q^k$, $e_2 \in R_q$ を、各成分多項式の係数が十分小さい値をとるよう選ぶ。次に, $c_1 = A^T \cdot r + e_1 \in R_q^k$, $c_2 = t^T \cdot r + e_2 + \lfloor q/2 \rfloor \cdot M \in R_q$ を計算し、組 (c_1, c_2) を暗号文とする。

■復号 暗号文 (c_1, c_2) に対し, $M' = c_2 - s^T \cdot c_1 \in R_q$ を計算する。 M' の各成分について, $\lfloor q/2 \rfloor$ に近ければ 1, そうでなければ 0 とすることで復号結果 $M'' \in R_q$ を得る。

3. 高速多項式乗算アルゴリズムToom-Cook法

Toom-Cook法とは、1963年にToomによって考案された、分割統治による再帰によって多項式積を高速に計算する方法である。以下、 K を体(四則演算が定義された集合)とし、 K の元を係数に持つ変数 x の多項式全体の集合を $K[x]$ とする。また、自然数 d, k をそれぞれ入力多項式の次数、多項式の分割数とし、 $d' = d/k$ とする。

(1)分割 入力多項式 $f, g \in K[x]$ を $f = \sum_{i=0}^{k-1} f_i(x)x^{di}$, $g = \sum_{i=0}^{k-1} g_i(x)x^{di}$ のように、次数 d' の部分多項式 $f_i, g_i \in K[x]$ を用いて表す。また、新たな変数 X に対し, $F = \sum_{i=0}^{k-1} f_i X^i$, $G = \sum_{i=0}^{k-1} g_i X^i$, $H = F \cdot G$ と定義する。

(2)評価 相異なる $2k-1$ 個の $\alpha_0, \dots, \alpha_{2k-2} \in K$ を選び、各 $0 \leq i \leq 2k-2$ に対し, d' 次多項式 $F(\alpha_i), G(\alpha_i) \in K[x]$ を計算後、積 $H(\alpha_i) = F(\alpha_i) \cdot G(\alpha_i)$ を再帰的に求める。

(3)補間・復元 $H(\alpha_i)$ から、Vandermonde行列の逆行列を用いて H を復元後, $f \cdot g = H(x^{d'})$ を得る。

また、漸近計算量は $O(d^{\log_k(2k-1)})$ と見積もられる。

4. Toom-Cook法のCRYSTALS-Kyberへの組込と実験

本研究では、CRYSTALS-KyberをPythonを用いて実装し、鍵生成、暗号化、復号に現れる多項式乗算部分をKaratsuba, Toom-3, Toom-4に置き換えた方式に対し、処理時間を比較した。パラメータはKyber512[2]に基づき、 $q = 3329$, $k = m = 2$ に固定し、 n を16から256まで16刻みで動かした(実際には n が 2 冪の時のみ使用する)。また、各 n に対し、鍵生成→暗号化→復号の一連処理を1000回繰り返して、各プロセスの処理時間を計測した。なお、実行環境は、OS: Windows 11 Pro 64bit, CPU: Intel(R) Core(TM) 7 150U (1.80 GHz), メモリ: 16GB, 開発環境: VScode である。

図1に各プロセスの平均処理時間の挙動を示す。

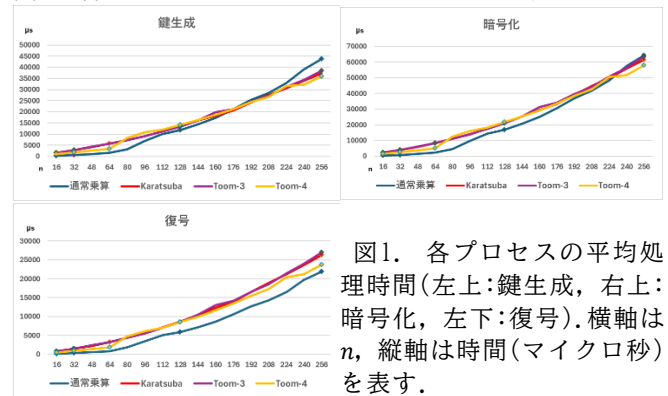


図1. 各プロセスの平均処理時間(左上:鍵生成, 右上:暗号化, 左下:復号).横軸は n , 縦軸は時間(マイクロ秒)を表す。

本実験の結果、160以下の n では通常乗算が最も効率的である。192以上の n では鍵生成、暗号化でKaratsuba, Toom-3, Toom-4が通常乗算を上回りはじめ、 $n = 256$ ではToom-4が最速となった。一方、復号では全体で通常乗算が最速となった。このことから、少なくとも鍵生成、暗号化ではToom-Cook法の分割数 k を増やせば増やすほどより大きい n で効率的になることが示唆される。

5. 今後の課題

今後の課題としては、処理時間だけではなくメモリ使用量も含めて比較を行いたい。具体的には、処理時間とメモリ使用量のトレードオフを考慮した上で、CRYSTALS-Kyberにおいて最適な分割数 k を理論的に決定したい。最適な k に対し、通常の乗算よりも高速となる n の閾値の決定も課題である。加えて、NTTとの比較や、NTTとToom-Cook法のハイブリッド、さらには、C++などのコンパイラ言語による実装についても検討する必要がある。

参考文献

- [1]廣澤佑亮, 松浦幹太: Ring-LWEにおける多項式乗算の高速化手法, 2025年 暗号と情報セキュリティシンポジウム(SCIS2025), 1A1-2.
[2] J. Bos et al.: CRYSTALS-Kyber: A CCA-Secure Module-Lattice-Based KEM. 2018 IEEE EuroS&P, London, UK, pp. 353-367, 2018.