Сравнение способов умножения в схемах Kyber + Lizard и NewHope + Lizard

К. А. Арканников*, Б. Г. Мубараков‡

Сравниваются три способа умножения полиномов в Kyber и NewHope отдельно и в схемах с неизменным Lizard. Приведены результаты и доверительные интервалы.

Ключевые слова: Kyber, NewHope, Lizard, Карацуба, NTT.

1. Введение и постановка задачи

Kyber и NewHope устанавливают общий ключ через открытый канал [1,2]. Lizard шифрует открытым ключом, расшифровывает секретным [3].

Цель работы — оценить ускорение Kyber и NewHope при замене обычного умножения полиномов алгоритмом Карацубы или теоретико-числовым преобразованием (NTT). Меняется только умножение; при одинаковых входах, параметрах и случайных последовательностях ключи и шифртексты сохраняются.

Lizard задаёт общую неизменную нагрузку стенда. Эффект модификаций выделяется отдельными замерами умножения; время составной схемы включает оба компонента.

Схема взаимодействия. Ключевые пары независимы. Отправитель получает s_P через Kyber/NewHope и передаёт 32-байтовый z через Lizard. Итоговый ключ объединяет s_P и полученный из z вклад s_L.

Для воспроизводимости зададим порядок байтов и метки объединения. $F(v) = \mathrm{LE}_8(|v|)\|v$ добавляет длину строки: восемь байтов, младший первым; $\|$ соединяет строки. Пусть $H = \text{SHA3-256}$, $X(v) = \text{SHAKE-256}(v, 32)$, c_P — шифртекст Kyber/NewHope, $|s_P| = 32$. Тогда

$$\begin{aligned} s_L &= X(D_L\|z), \qquad K = X(T),\\ T &= F(D)\|F(I)\|F(s_P)\|F(s_L)\\ &\quad \|F(H(c_P))\|F(H(E_L))\|F(C). \end{aligned}$$

Арканников Кирилл Артурович — магистрант каф. САИТ, ИВМиИТ КФУ, e-mail: kaarkannikov@kpfu.ru.

Arkannikov Kirill Arturovich — master's student, Department of System Analysis and Information Technologies, Kazan Federal University.

‡*Мубараков Булат Газинурович* — канд. техн. наук, доцент каф. САИТ, ИВМиИТ КФУ, e-mail: mubbulat@mail.ru, ORCID: 0000-0002-7628-9870.

Mubarakov Bulat Gazinurovich — docent, Kazan Federal University.

Опубликовано на условиях лицензии CC-BY 4.0
Published under CC-BY 4.0

В ASCII заданы D =`KYBER-NEWHOPE-LIZARD-HYBRID-v1`, D_L =`LIZARD-HYBRID-CONTRIBUTION-v1` и фиксированная метка эксперимента C =`master-thesis-hybrid-pair-benchmark-v1`. Название I состоит из `Kyber`/`NewHope`, суффикса `-Karatsuba`, `-NTT` либо пустой строки и `+Lizard`. Совпадение K проверяется у сторон внутри каждого варианта.

Шифртекст Lizard кодируется как $E_L = D_c \| F(c_1) \| F(V_p(u)) \| F(V_p(v)) \| F(V_t(w))$, где D_c =`LIZARD-CCA-CIPHERTEXT-v1`. Функция V_a записывает коэффициенты по порядку, по модулю a, по $\lceil \log_2(a)/8 \rceil$ байтов, младший первым. Длина c_1 — 32 байта, u, v, w — по 256 коэффициентов. Получатель восстанавливает s_P, z и повторяет вычисления.

2. Параметры и модификации

В Kyber $n = 256, q = 3329, k = 4$, в NewHope $n = 1024, q = 12289$: n — число коэффициентов, q — модуль, k — длина вектора Kyber. Произведения сокращаются по $x^n + 1$ и q. Алгоритм Карацубы разбивает каждый полином на две половины: вместо четырёх произведений половин достаточно трёх [4]; подполиномы, содержащие не более 32 коэффициентов, умножаются обычным способом.

В NTT [1,2] при $n = Md$, $a(x) = \sum_{r=0}^{d-1} x^r a_r(x^d)$ вычисляются значения a_r в точках $\beta_j = \psi^{2j+1}$, $0 \le j < M$, с порядком ψ, равным $2M$. Блоки перемножаются по модулю $x^d - \beta_j$, затем выполняется обратное NTT. В Kyber неполное NTT даёт для каждого полинома 128 пар коэффициентов остатков $a_{j,0} + a_{j,1}x$, в NewHope — 1024 скаляра. При $n = 1024, q = 3329$ блоков 128 по 8 коэффициентов, порядок $\psi = 3061$ равен 256, не 2048.

В Lizard $(m, n, q, p, t, \ell) = (768, 256, 3329, 1024, 2, 256)$, $\sigma = 3329/(182\sqrt{2\pi})$. Открытый ключ — матрицы $m \times n$ и $m \times \ell$ по модулю q, секретный — $n \times \ell$: $P(0) = 1/2$, $P(1) = P(-1) = 1/4$. Элементы A равномерны по модулю q. Параметр t задаёт алфавит $\{0, \ldots, t-1\}$; при $t = 2$ сообщение — $\ell = 256$ равномерных битов. Для ошибок $P(E = x) \propto e^{-\pi x^2/\sigma^2}$ на $[-44, 44]$. Суммы $F_E(x) = P(E \le x)$ вычисляются с плавающей точкой. Для случайного 64-битного целого R двоичный поиск выбирает минимальное x с $F_E(x) \ge R/2^{64}$. Обёртка выводит из SHAKE-256 вектор со 128 ненулевыми знаками и проверяет повторное шифрование и контрольный вектор [3].

Ключи Lizard — списки Python без упаковки. Оценки размеров при 12 и 2 битах на коэффициент: $2mn \cdot 12/8 = 589824$ и $n\ell \cdot 2/8 = 16384$ байта; это не расход памяти. Модули фиксированы, тесты простоты [5] не измеряются.

3. Измерения и результаты

Эксперимент 18.10.2026: Ryzen 7 7800X3D, Windows 11 (26200), CPython 3.13.0; три прогрева, 30 пар замеров в случайном порядке, 10000 повторных выборок (seed=123456). Генераторы воспроизводимы; сборщик мусора отключён. Ключи и таблицы NTT подготовлены вне замеров; пять групп тестов пройдены.

В таблице Ф/В — формирование/восстановление ключа, t_K, t_N — медианы времени. Интервал оценивает медиану $E_i = 100(1 - t_{K,i}/t_{N,i})$: при $E_i > 0$ быстрее Kyber-пара, при $E_i < 0$ — NewHope-пара.

Метод		t_K, мс	t_N, мс	95% интервал E, %
Обычный	Ф	113,804	272,078	[57,1; 59,3]
Обычный	В	125,007	381,177	[66,8; 67,7]
Карацуба	Ф	46,112	45,325	[−2,7; −0,9]
Карацуба	В	57,229	69,454	[17,1; 17,9]
NTT	Ф	19,601	13,446	[−48,2; −44,9]
NTT	В	27,122	22,226	[−24,1; −20,3]

При обычном умножении быстрее Kyber-пара, при NTT — NewHope-пара. Знаковый тест без поправки на множественные сравнения даёт $p = 0{,}0003249$ для Карацубы/Ф и $p = 1{,}863 \cdot 10^{-9}$ иначе.

При $q = 3329$, $n = 256, 1024$ умножение ускорилось в 2,52–6,33 раза с Карацубой и в 7,95–33,97 с NTT (медианы парных отношений), включая эффект изменения частоты приведения по модулю.

Выводы и перспективы. В эксперименте обе модификации ускорили умножение. Далее сравним инкапсуляцию и декапсуляцию до и после модификации. Замеры на данном коде и компьютере не доказывают безопасность модификаций и объединения ключей, равную стойкость или соответствие FIPS 203.

Список литературы

[1] NIST, *Module-Lattice-Based Key-Encapsulation Mechanism Standard. FIPS 203*, 2024. DOI: 10.6028/NIST.FIPS.203.

[2] E. Alkim et al., *NewHope: Algorithm Specifications and Supporting Documentation. Version 1.1*, 2020, https://newhopecrypto.org/data/NewHope_2020_04_10.pdf.

[3] J. H. Cheon, D. Kim, J. Lee, Y. Song, *Lizard: Cut off the Tail! Practical Post-Quantum Public-Key Encryption from LWE and LWR*, Cryptology ePrint Archive, 2016/1126, 2016, https://eprint.iacr.org/2016/1126.

[4] А. Карацуба, Ю. Офман, “Умножение многозначных чисел на автоматах”, *Докл. АН СССР*, **145**:2 (1962), 293–294.

[5] Ш. Т. Ишмухаметов, Н. А. Антонов, Б. Г. Мубараков, Р. Г. Рубцова, “Об одном комбинированном тесте простоты”, *Изв. вузов. Матем.*, 2022, № 12, 123–129. DOI: 10.26907/0021-3446-2022-12-123-129 .

Статья поступила 9 октября 2026 г.

Comparing Multiplication Methods in Kyber + Lizard and NewHope + Lizard Schemes

K. A. Arkannikov, B. G. Mubarakov

Three polynomial multiplication methods in Kyber and NewHope are compared separately and in schemes with unchanged Lizard. Results and confidence intervals are reported.

Keywords: Kyber, NewHope, Lizard, Karatsuba, NTT.

References

[1] NIST, *Module-Lattice-Based Key-Encapsulation Mechanism Standard. FIPS 203*, 2024. DOI: 10.6028/NIST.FIPS.203.

[2] E. Alkim et al., *NewHope: Algorithm Specifications and Supporting Documentation. Version 1.1*, 2020, https://newhopecrypto.org/data/NewHope_2020_04_10.pdf.

[3] J. H. Cheon, D. Kim, J. Lee, Y. Song, *Lizard: Cut off the Tail! Practical Post-Quantum Public-Key Encryption from LWE and LWR*, Cryptology ePrint Archive, 2016/1126, 2016, https://eprint.iacr.org/2016/1126.

[4] A. Karatsuba, Yu. Ofman, “Multiplication of many-digital numbers by automatic computers”, *Dokl. Akad. Nauk SSSR*, **145**:2 (1962), 293–294 (In Russian).

[5] Sh. T. Ishmukhametov, N. A. Antonov, B. G. Mubarakov, R. G. Rubtsova, “On a Combined Primality Test”, *Izv. Vyssh. Uchebn. Zaved. Mat.*, 2022, № 12, 123–129. DOI: 10.26907/0021-3446-2022-12-123-129 (In Russian).

Received on October 9, 2026