УДК 004.056
doi: 10.26583/bit.2024.4.06

Игорь Ю. Жуков[1], Виталий Г. Иваненко[2], Ирина Д. Иванова[3], Нина Д. Иванова[4]
[1,2]Национальный исследовательский ядерный университет «МИФИ»,
Каширское ш., 31, Москва, 115409, Россия
[3,4]Российский университет транспорта (МИИТ),
ул. Образцова, 9, стр. 9, Москва, 127994, Россия
[1]e-mail: i.zhukov@inbox.ru, https://orcid.org/0000-0002-4429-8799
[2]e-mail: vgivanenko@mephi.ru, https://orcid.org/0000-0003-0823-5501
[3]e-mail: iid.ivanova@yandex.ru, https://orcid.org/0000-0003-3022-8973
[4]e-mail: ivanovand.nina@yandex.ru, https://orcid.org/0000-0001-5942-8050

УСКОРЕНИЕ МОДУЛЬНОЙ АРИФМЕТИКИ В ПОСТКВАНТОВЫХ СХЕМАХ ПОДПИСИ

Аннотация. С момента открытия квантового алгоритма Шора, способного за полиномиальное время решить задачи факторизации и дискретного логарифмирования, в мире ведется активная разработка постквантовых криптографических алгоритмов. Хотя некоторые из постквантовых алгоритмов уже стандартизированы NIST, важной проблемой остается их меньшая эффективность в сравнении с классическими криптографическими алгоритмами, используемыми в настоящее время. Целью данной работы является ускорение операций над полиномами в постквантовых схемах подписи, основанных на теории решеток. Исследование проводится путем анализа быстрых алгоритмов умножения полиномов и алгоритмов приведения чисел по модулю. В работе рассмотрены числовое теоретическое преобразование (NTT), применяемое для ускорения умножения полиномов в конечных полях, и используемые им алгоритмы приведения по модулю K-RED и Монтгомери. Обоснована эффективность алгоритма K-RED в сравнении с алгоритмом Монтгомери, используемым в эталонных реализациях постквантовых схем подписи. Рассмотрен вопрос применимости преобразования NTT и быстрых алгоритмов приведения чисел по модулю для внедрения в состав российской схемы подписи «Крыжовник» и американских алгоритмов Falcon и CRYSTALS-Dilithium. В результате исследования приведены рекомендации по встраиванию алгоритма K-RED в эталонные реализации схем подписи Falcon и CRYSTALS-Dilithium, а также осуществлена реализация на языке Си ускоренного умножения полиномов в конечных полях при помощи преобразования NTT и алгоритма K-RED. Результаты настоящего исследования могут быть внедрены в стандарты NIST постквантовых схем подписи FIPS 204 и FIPS 206, а также адаптированы для использования в иных схемах подписи, основанных на теории решеток и использующих факторкольца.

Ключевые слова: теория решеток, постквантовая криптография, приведение числа по модулю, умножение полиномов, факторкольцо.
Для цитирования: ЖУКОВ, Игорь Ю. и др. УСКОРЕНИЕ МОДУЛЬНОЙ АРИФМЕТИКИ В ПОСТКВАНТОВЫХ СХЕМАХ ПОДПИСИ. Безопасность информационных технологий, [S.l.], т. 31, № 4, с. 99–108, 2024. ISSN 2074-7136. URL: https://bit.spels.ru/index.php/bit/article/view/1718. DOI: http://dx.doi.org/10.26583/bit.2024.4.06.

Igor Y. Zhukov[1], Vitaliy G. Ivanenko[2], Irina D. Ivanova[3], Nina D. Ivanova[4]
[1,2]National Research Nuclear University MEPhI (Moscow Engineering Physics Institute),
Kashirskoe shosse, 31, Moscow, 115409, Russia
[3,4]Russian University of Transport (MIIT),
Obraztsova str., 9, p. 9, Moscow, 127994, Russia
[1]e-mail: i.zhukov@inbox.ru, https://orcid.org/0000-0002-4429-8799
[2]e-mail: vgivanenko@mephi.ru, https://orcid.org/0000-0003-0823-5501
[3]e-mail: iid.ivanova@yandex.ru, https://orcid.org/0000-0003-3022-8973

[4]*e-mail: ivanovand.nina@yandex.ru, https://orcid.org/0000-0001-5942-8050*

Acceleration of modular arithmetic in post-quantum signature schemes

Abstract. Since the discovery of the Shor quantum algorithm, capable of solving factorization and discrete logarithm problems in polynomial time, post-quantum cryptographic algorithms have been actively developed. Although some of the post-quantum algorithms have already been standardized by NIST, an important problem remains their lower efficiency compared to the classical cryptographic algorithms. The purpose of this work is to accelerate operations on polynomials in lattice-based post-quantum signature schemes. The research is carried out by analyzing fast multiplucation and reduction algorithms. The paper considers the Number Theoretic Transform (NTT), K-RED and Montgomery reduction algorithms. The effectiveness of the K-RED algorithm in comparison with the algorithm Montgomery is substantiated. The question of the applicability of the NTT and fast reduction algorithms in the Russian signature scheme «Kryzhovnik» and NIST algorithms Falcon and CRYSTALS-Dilithium is considered. As a result of the research, recommendations for embedding the K-RED algorithm in the reference implementations of the Falcon and CRYSTALS-Dilithium signature schemes are given, as well as the implementation in C of accelerated multiplication using the NTT and the K-RED algorithm. The results of this research can be embedded in the NIST standards FIPS 204 and FIPS 206, as well as adapted for use in other lattice-based signature schemes.

Keywords: lattices, post-quantum cryptography, reduction algorithm, multiplication of polynomials, quotient ring.

For citation: ZHUKOV, Igor Y. et al. Acceleration of modular arithmetic in post-quantum signature schemes. IT Security (Russia), [S.l.], v. 31, no. 4, p. 99–108, 2024. ISSN 2074-7136. URL: https://bit.spels.ru/index.php/bit/article/view/1718. DOI: http://dx.doi.org/10.26583/bit.2024.4.06.

Введение

В настоящее время одним из наиболее перспективных подходов к построению постквантовых схем шифрования и подписи является криптография, основанная на теории решеток. Среди финалистов конкурса NIST и российских проектов постквантовых схем подписей можно выделить следующие алгоритмы, использующие решетки:

Алгоритм Falcon, являющийся финалистом конкурса NIST среди постквантовых схем подписи, основывается на NTRU-решетках и парадигме «хеширование и подпись». Данный алгоритм сложен в реализации за счет использования древовидной структуры хранения данных (LDL дерева) при создании подписей, обширных операций с плавающей запятой (применяющихся при создании подписи и частично в процессе генерации ключей) и нормального распределения (вызываемого для каждого листа LDL дерева). Наиболее дорогостоящей является операция генерации ключей, однако за счет малых коммуникационных затрат [1] Falcon считается сильным кандидатом на внедрение в маломощные устройства.

Алгоритм CRYSTALS-Dilithium, ставший финалистом конкурса NIST, основывается на парадигме Фиата-Шамира с прерываниями и задаче LWE (Learning With Errors, обучение с ошибками). В отличие от Falcon, в CRYSTALS-Dilithium работа ведется лишь с факторкольцом $\mathbb{Z}_q[x]/(x^n+1)$, что позволяет избежать операций с плавающей запятой. В процессе генерации подписей используется равномерное распределение, также CRYSTALS-Dilithium демонстрирует лучшие, нежели Falcon, показатели в скорости выполнения [2]. Тем не менее из-за того, что подпись CRYSTALS-Dilithium состоит из двух матриц, коммуникационные затраты алгоритма являются значительными.

Алгоритм «Крыжовник», предлагаемый в качестве проекта российского стандарта постквантовой схемы подписи, основывается на парадигме Фиата-Шамира с прерываниями и задаче LWR (Learning With Rounding, обучение с округлением), причем

требуемая для нее матрица A должна удовлетворять условию, чтобы хотя бы один из полиномов в ее составе являлся обратимым [3]. Операции проводятся в факторкольце и используют равномерное распределение. При параметрах, соответствующих одинаковому уровню стойкости, алгоритм «Крыжовник» выполняется медленнее в сравнении с CRYSTALS-Dilithium и требует больших коммуникационных затрат.

С вычислительной точки зрения фундаментальной и к тому же трудоемкой операцией во многих схемах, основанных на стандартных решетках, является умножение векторов над конечным полем, тогда как в рассматриваемых схемах умножение проводится над элементами кольца многочленов, для которых необходимо одновременно учитывать особенности операций в конечных полях и структуру факторкольца. В данной работе предлагается метод синтеза быстрых алгоритмов умножения полиномов и алгоритмов приведения чисел по модулю, с помощью которого в постквантовых схемах подписи на основе решеток возможно ускорить операции генерации ключей, создания и проверки подписей. Также рассматривается применимость предлагаемого метода для внедрения в перспективные постквантовые алгоритмы.

1. Алгоритмы умножения полиномов в факторкольце

Описанные постквантовые схемы подписи основываются на модификациях задачи SIS (Short Integer Solution, короткое целочисленное решение), которая в общем случае может быть сформулирована следующим образом [4]:

$$\begin{cases} \|\vec{x}\| \leq \beta \\ f_A(\vec{x}) := A\vec{x} = \vec{0} \in \mathbb{Z}_q^n, \end{cases} \quad (1)$$

где матрица $A = [a_1] \dots [a_m]$ состоит из случайных векторов из $\mathbb{Z}_q^n$, а $\vec{x}$ – ненулевой вектор из $\mathbb{Z}_q^m$, который необходимо найти.

При этом в качестве начальных данных для задачи SIS на вход подаются вектора, чьи коэффициенты являются коэффициентами полиномов, определяемых в факторкольце многочленов над конечным полем. Для упрощения и ускорения операций умножения над полиномами в криптографии применяются специализированные алгоритмы [5]. Наиболее распространенными алгоритмами умножения полиномов являются алгоритм Карацубы, алгоритм Тоома-Кука (обобщение алгоритма Карацубы), алгоритм БПФ и алгоритм NTT (вариант БПФ для конечных полей).

С точки зрения реализации, наиболее эффективными являются БПФ и NTT (с использованием «бабочек» Кули-Тьюки и Джентельмена-Санде) за счет сложности $O(n \log n)$, что обуславливает их значительное преимущество перед всеми другими алгоритмами умножения. Отличие между БПФ и NTT заключается в том, что в БПФ вычисления выполняются в комплексном поле, в то время как в NTT работа ведется с целыми числами, что позволяет избежать ошибок точности округления при умножении двух полиномов с целочисленными коэффициентами.

Помимо этого у NTT есть ряд других преимуществ [6]:

1. Линейность прямого и обратного преобразований NTT.

2. Сохранение случайности на выходе преобразования при передаче на вход случайного полинома. Вследствие этого, если необходимо вычислить произведение полинома a и случайного полинома b, вместо нахождения прямого преобразования NTT для первого полинома $\tilde{a} = NTT(a)$ можно принять $\tilde{a} = a$.

3. Сохранение размерности и битовой длины. За счет данного свойства результирующий полином $\tilde{a}$ может размещаться там же, где была выделена память под исходный полином a.

4. В тех случаях, когда полином a задействуется в нескольких умножениях, удобно единожды вычислить $\tilde{a}$ и хранить полученное значение для дальнейших вычислений без расхода дополнительной памяти.

5. В то же время для применения преобразования NTT необходимо, чтобы операции проводились над полем по модулю простого числа, а работа в факторкольце $\mathbb{Z}_q[x]/(x^n+1)$ требует к тому же существования ψ, корня $2n$-ой степени из единицы, для которого бы выполнялось:

$$\psi^{2n} = 1 \, mod \, q. \tag{2}$$

2. Алгоритмы приведения по модулю числа в факторкольце

Арифметика в преобразовании NTT требует большого количества операций сложения, вычитания и умножения по модулю целого числа [7]. Хотя во многих языках программирования предусмотрена операция взятия остатка от деления на целое число, в большинстве случаев подход, при котором за организацию процесса нахождения отвечает реализация этой операции в конкретном языке программирования, является неудовлетворительным.

Форма (или область) Монтгомери – другой способ выражения элементов кольца, позволяющий свести приведение по модулю к операциям целочисленного умножения и битового сдвига. В отличие от деления, данные операции являются оптимизированными и обычно выполняются за фиксированное число тактов, не зависящее от обрабатываемых данных.

Использование алгоритма Монтгомери требует приведения одного из множителей в форму Монтгомери. Поскольку операция перевода числа в форму Монтгомери довольно ресурсоемкая, на практике стараются выбирать для этого числа, которые в ходе проведения вычислений могут быть использованы несколько раз, иначе алгоритм Монтгомери может быть достаточно дорогостоящим. Очевидно, что в преобразовании NTT с этой целью могут использоваться коэффициенты поворота, которые одинаковы для всех полиномов в выбранном кольце $\mathbb{Z}_q$.

В отношении модулей, что могут быть представлены как:

$$q = k * 2^m + 1, \tag{3}$$

где k, m — целые числа и $k \geq 3$ мало, в качестве алгоритма приведения может быть использован алгоритм K-RED [8]. На выходе данный алгоритм выводит значение, которое пропорционально входному значению, чем напоминает форму Монтгомери, применяемую в алгоритме Монтгомери. Однако в данном случае произведение «масштабируется» не с помощью выбранного r, а посредством k, получаемого в результате представления порядка q в виде (3).

Представим приводимое число как $c = c_0 + 2^m * c_1$, $0 \leq c_0 < 2^m$. Тогда алгоритм K-RED для входного числа c и известных k и m может быть записан следующим образом:

1. Присваиваем $c_0 = c \, mod \, 2^m$ и $c_1 = \left\lfloor \frac{c}{2^m} \right\rfloor$.

2. Возвращаем $c' = kc_0 - c_1$.

Хотя описанный алгоритм не возвращает число в пределах $0 \leq c' < q$, для малого k путем прибавления либо вычитания нескольких кратных q множителей можно получить необходимое приведенное значение. В случае вычисления NTT для большого n во избежание переполнений также может потребоваться еще большая редукция некоторых из промежуточных значений, для чего может использоваться алгоритм K-RED-2x.

Аналогично алгоритму Монтгомери, алгоритм K-RED требует предварительного масштабирования коэффициентов поворота. В общем случае они могут быть рассчитаны по алгоритму 1:

Алгоритм 1 – Масштабирование коэффициентов поворота прямого преобразования NTT

*g_k = g * k^-2*
g_k = K-RED(g_k)
x = k^-1 mod p
for (i = 0; i < n; i ++) do
    v = f(REV, i)
    ψ[v] = x
    *x = K-RED(x * g_k)*
end for

В данном алгоритме g – корень n-ой степени из единицы, k – параметр алгоритма K-RED, а элементы массива ψ составляют коэффициенты поворота, применяемые в прямом NTT. Поскольку в результате выполнения алгоритма K-RED на выход подается приведенное значение для $k * c$ вместо исходного c, при вычислении масштабированных коэффициентов поворота необходимо, чтобы оба множителя были масштабированы на множитель k^{-1}.

Следует отметить, что ввиду того, что в алгоритме Кули-Тьюки входные данные подаются в обратном битовом порядке [9], фактический индекс v, по которому происходит обращение к элементам таблицы в алгоритме выше, вычисляется при помощи предварительно вычисленного массива *REV*. Таким образом, значение индекса v является выходом функции, зависящей от текущей итерации и массива обратных индексов.

3. Применимость преобразования NTT к алгоритму «Крыжовник»

В схеме подписи «Крыжовник» вычисления проводятся в кольце многочленов $\mathbb{Z}_q[x]/(x^n + 1)$, где модуль $q = 2^v$. Данный выбор модуля обеспечивает защиту от атак, использующих кольцевую структуру задачи LWR, и большую гибкость при повышении уровня безопасности, тем самым позволяя осуществлять переход на большие модули без изменения внутренней арифметики [10]. Вследствие данной особенности при умножении полиномов в данном факторкольце невозможно применить преобразование NTT. Однако ввиду того, что «Крыжовник» использует полиномы сравнительно небольшой длины ($n = 256$), применение общих алгоритмов умножения не приводит к значительному замедлению работы схемы подписи. В эталонной реализации схемы подписи «Крыжовник» приведены на выбор несколько таких алгоритмов: «школьный метод» и гибридный метод, предложенный в постквантовом алгоритме SABER.

Однако и при применении общих алгоритмов умножения полиномов требуется сохранять результаты вычислений в кольце $\mathbb{Z}_q$, для чего используются алгоритмы приведения по модулю. В отличие от криптосхем, в составе которых применяются модули вида $q \equiv 1 \; mod \; n$, в схеме подписи «Крыжовник» нахождение остатка от деления на 2^v может быть сведено к аппаратному отсечению нужного количества младших битов. Так, в то время как в эталонной реализации на языке Си схемы подписи «Крыжовник» для выполнения операций в кольце используется библиотека NTL, разработанная специально для выражения объектов теории чисел, в эталонной реализации постквантового механизма шифрования ключей (Key Encapsulation Mechanism, KEM) SABER данное приведение по модулю реализовано следующим образом:

```
val & (moduli - 1),
```

где `val` – приводимое по модулю значение, `moduli` – применяемый модуль, а операция `&` – побитовое «И».

Таким образом, хотя из-за выбора модуля в схеме подписи «Крыжовник» невозможно использовать умножение полиномов, построенное на преобразовании NTT, оно позволяет наиболее просто осуществлять приведение по модулю без необходимости перехода на альтернативные модули, как в случае алгоритма Монтгомери. При этом применение алгоритма K-RED в схеме подписи является невозможным из-за вида модуля (3).

4. Ресурсоемкие операции в Falcon

Схема подписи Falcon основывается на NTRU-решетках, характеризующихся тем, что при размерности пространства $n = 2^l$ закрытый ключ системы является кортежем полиномов (f, g, F, G). Данные полиномы определены в факторкольце по модулю $x^n + 1$ с малыми коэффициентами в поле $\mathbb{Z}_q$. При условии, что полиномы f и g выбираются при помощи соответствующего закона распределения (нормального распределения для 1-го и 5-го уровней стойкости в алгоритме Falcon), полиномы F и G должны удовлетворять уравнению NTRU:

$$fG - gF = q \ mod \ (x^n + 1). \quad (4)$$

В эталонной реализации схемы Falcon поиск решения осуществляется рекурсивно, что позволяет минимизировать код, однако это требует дополнительной памяти для хранения родительских функций и аргументов. Поиск решения уравнения NTRU является наиболее дорогостоящим этапом в процессе генерации ключей. В известных нам исследованиях на тему его ускорения предлагается применять более сложные в реализации матричные операции вместо рекурсивных [11], либо внедрять в рекурсию улучшенные техники, касающиеся сохранения промежуточных значений полиномов f и g вместо их повторного вычисления и уменьшения потребления памяти на 0-м и 1-м уровне рекурсии [12]. Для обоих подходов все еще актуальной является проблема приведения по модулю произведения полиномов, потому в данной работе нами рассматривается возможность ускорения данной операции.

Поиск решения уравнения NTRU в алгоритме Falcon значительно упрощает применение преобразования NTT. При этом используется система остаточных классов [13], в которой значение переменной x определяет систему равенств вида:

$$x_i \equiv x \ mod \ p_i, \quad (5)$$

где $\{p_i\}$ — список различных простых модулей p_i.

В эталонной реализации Falcon эти модули подбираются таким образом, чтобы каждый из них был немногим меньше 2^{31} (чтобы упростить работу с целочисленной арифметикой) и удовлетворял соотношению:

$$p_i \equiv 1 \ mod \ 2n, \quad (6)$$

где n – длина полинома. Выполнение данного соотношения необходимо, чтобы при умножении полиномов было возможно применить преобразование NTT.

Однако в ходе решения уравнения NTRU выбор модуля p, по которому рассчитываются коэффициенты преобразования NTT, не является фиксированным, поэтому в `keygen.c`, реализующем алгоритм генерации ключей, коэффициенты поворота в форме Монтгомери для соответствующего модуля вычисляются непосредственно в процессе выполнения алгоритма. Поскольку новые коэффициенты поворота получаются путем домножения младших по индексу коэффициентов на корень

из единицы, данный процесс также требует приведения произведения по модулю. Для этого в эталонной реализации также применяется алгоритм Монтгомери.

5. Ресурсоемкие операции в CRYSTALS-Dilithium

В схеме подписи CRYSTALS-Dilithium вычисления проводятся в факторкольце для $n = 256$. Для применения NTT в отрицательно завернутых свертках в CRYSTALS-Dilithium аналогично Falcon модуль q подбирается таким образом, чтобы выполнялось соотношение:

$$q \equiv 1 \; mod \; 2n. \tag{7}$$

С этой целью в данной схеме подписи используется модуль $q = 2^{23} - 2^{13} + 1$, а корень из единицы соответственно равен константе $\omega = 1753$. За счет того, что модуль поля является фиксированной величиной, таблица масштабированных коэффициентов поворота может быть рассчитана заранее по описанному ранее алгоритму 1.

При этом приведенное по модулю q значение величины a в соответствии со спецификацией удовлетворяет соотношениям:

$$\begin{gathered} -\frac{q}{2} < a' \leq \frac{q}{2}, \\ -\frac{q-1}{2} \leq a' \leq \frac{q-1}{2}, \end{gathered} \tag{8}$$

для четного и нечетного значения a соответственно. Авторы CRYSTALS-Dilithium и CRYSTALS-Kyber такое приведение по модулю называют «отцентрованным приведением» (Centered Reduction), и именно оно используется при вычислении таблицы масштабированных коэффициентов поворота в преобразовании NTT.

В эталонной реализации CRYSTALS-Dilithium таблица коэффициентов поворота хранится в массиве *zetas*, и значения в ней приведены по модулю q в соответствии с описанным отцентрованным приведением. Таким образом, среди коэффициентов поворота в схеме подписи CRYSTALS-Dilithium допускаются отрицательные величины, которые при умножении на положительные коэффициенты полиномов дадут в произведении отрицательные коэффициенты в форме NTT. Оценки алгоритма K-RED работают для неотрицательных значений, потому, чтобы внедрить алгоритм K-RED в схему подписи CRYSTAL-Dilithium, необходимо пересчитать таблицу коэффициентов поворота в соответствии с диапазоном $[0, q)$. При этом, если выходное значение K-RED необходимо привести к «отцентрованной» форме в соответствии с соотношениями (8), на практике это легко реализуется за счет добавления еще одного условного выражения. Однако в модуле `poly.c` эталонной реализации CRYSTALS-Dilithium на выходное значение INTT накладывается лишь ограничение, чтобы оно по абсолютной величине не превышало q, вследствие чего возможно применение классического алгоритма K-RED.

6. Ускорение схем подписей, основанных на теории решеток

В результате анализа спецификаций упомянутых схем подписи обнаружено, что в тех криптографических схемах, где возможно использование преобразования NTT для ускорения операций умножения полиномов, в качестве алгоритма приведения предлагается использовать алгоритм Монтгомери. При этом исследования, посвященные сравнению производительности алгоритмов приведения, показывают, что алгоритм K-RED позволяет ускорить выполнение преобразования NTT в 2 раза в сравнении с алгоритмом Монтгомери, а в составе криптографических схем дает выигрыш в производительности до 10% [14, 15].

Вид модулей, используемых при решении уравнения NTRU в схеме подписи Falcon, позволяет применить в преобразовании NTT алгоритм приведения K-RED. В выполненной реализации на языке Си предвычисленные по алгоритму 1 таблицы масштабированных коэффициентов для прямого и обратного преобразований NTT сохраняются в массивах KTF (K-RED Twiddle Factors) и `iKTF` соответственно. В функцию `modp_NTT2_ext`, при этом внесены следующие изменения:

1) Применение алгоритма K-RED требует предварительного вычисления таблицы масштабированных коэффициентов поворота. После нахождения элементов данной таблицы по алгоритму 1 обращение к ним осуществляется по индексам внешнего и среднего циклов алгоритма Кули-Тьюки.

```
s = KTF[m + u];
```

2) Поскольку в зависимости от итерации внешнего цикла применяется либо алгоритм K-RED, либо K-RED-2x, во внутреннем цикле необходимо добавить условие для приведения переменных по модулю.

```
if (m == 128) {
     u = KRED(u);
     v = KRED2x(v); }
else
     v = KRED(v);
```

3) Ввиду того, что при использовании алгоритма K-RED приведение значений осуществляется лишь при умножении чисел, при вычислении коэффициентов по «бабочкам» Кули-Тьюки достаточно применить целочисленные операции сложения и вычитания.

Помимо схожих изменений, в функции `mq_iNTT` необходимо осуществить корректировку результирующих значений при помощи предвычисленных констант. Для этого границы внешнего цикла алгоритма Джентльмена-Санде, применяемого для ускорения обратного преобразования NTT, смещаются с $m > 1$ до $m > 2$, и в отдельном цикле, где в классическом варианте INTT происходит масштабирование коэффициентов обратного преобразования на множитель n^{-1}, рассчитываются бабочки последнего уровня одновременно с их масштабированием по следующему алгоритму:

```
for (j = 0; j < n/2; j ++) do
     u = a[j];
     v = a[j+t];
     a[j] = KRED((u + v) * n0IK);
     a[j + t] = KRED((u - v) * s0IK);
end for
```

В данном алгоритме $n0IK = n^{-1} * k^{-s}$, а $s0IK = n^{-1} * k^{-s+1} * \psi^{-1}[1]$ (где ψ^{-1} – массив коэффициентов поворота в обратном преобразовании NTT). При этом показатель s зависит от 1) числа применений алгоритма K-RED-2x, 2) числа операций покомпонентного умножения, совершенных в результате проведения вычислений над полиномами в форме NTT. Также необходимо учитывать, что финальное масштабирование коэффициентов также проводится при помощи алгоритма K-RED, что увеличивает значение показателя s на 1.

Заключение

В результате данной работы сформированы рекомендации к ускорению работы постквантовых схем подписи путем применения быстрых алгоритмов в процессе

умножения полиномов. Выявлено, что в рассматриваемых схемах подписи Falcon и CRYSTALS-Dilithium вычисления проводятся в факторкольцах по модулю простого числа, что позволяет ускорить проводимые ими операции умножения за счет применения преобразования NTT и алгоритма приведения чисел по модулю K-RED. Определены особенности внедрения алгоритма K-RED в состав Falcon и CRYSTALS-Dilithium и с их учетом на языке программирования Си осуществлена реализация модулей, вычисляющих преобразование NTT с использованием алгоритма K-RED.

Также в ходе работы выявлено, что поскольку российской постквантовой схемой подписи «Крыжовник» в качестве модуля факторкольца выбрано число, являющееся степенью 2, в отношении нее не может быть применено ни преобразование NTT, ни алгоритм приведения чисел по модулю K-RED. Тем не менее предложенный в данной работе метод ускорения операций над полиномами может быть внедрен в те постквантовые алгоритмы, которые используют факторкольца по модулю простого числа. Среди них – стандарты NIST постквантовых схем подписи FIPS 204 и FIPS 206, основанные на алгоритмах CRYSTALS-Dilithium и Falcon соответственно.

В дальнейших исследованиях планируется оценить ускорение операций генерации ключей, создание и проверки подписей в алгоритмах Falcon и CRYSTALS-Dilithium, достигаемое за счет внедрения в них алгоритма K-RED. Также возможным направлением является внедрение модифицированных постквантовых схем в состав протокола TLS и оценка его быстродействия в сравнении с версиями, использующими классические криптографические алгоритмы.

СПИСОК ЛИТЕРАТУРЫ:

1. Allende M. et al. Quantum-resistance in blockchain networks. Scientific Reports. 2023, v. 13, no. 1, p. 5664. DOI: http://dx.doi.org/10.1038/s41598-023-32701-6.
2. Raavi M. et al. Security Comparisons and Performance Analyses of Post-Quantum Signature Algorithms. Lecture Notes in Computer Science. 2021, v. 12727, 24 p. DOI: http://dx.doi.org/10.1007/978-3-030-78375-4_17.
3. Киршанова Е.А. и др. Проект стандартизации постквантовой цифровой подписи. Прикладная дискретная математика. Приложение. 2020, № 13, с. 44–51. DOI: http://dx.doi.org/10.17223/2226308X/13/14.
4. Ducas L. et al. Finding short integer solutions when the modulus is small. Annual International Cryptology Conference. 2023, p. 150–176. DOI: http://dx.doi.org/10.1007/978-3-031-38548-3_6.
5. Alkim E. et al. Polynomial Multiplication in NTRU Prime: Comparison of Optimization Strategies on Cortex-M4. (2020). IACR Transactions on Cryptographic Hardware and Embedded Systems, 2021(1), p. 217–238. DOI: http://dx.doi.org/10.46586/tches.v2021.i1.217-238.
6. Liang Z., Zhao Y. Number theoretic transform and its applications in lattice-based cryptosystems: A survey. arXiv preprint arXiv:2211.13546. 2022. – 35 p. DOI: http://dx.doi.org/10.48550/arXiv.2211.13546.
7. Becker H. et al. Polynomial multiplication on embedded vector architectures. Cryptology ePrint Archive. 2021. – 29 p. URL: https://eprint.iacr.org/2021/998 (дата обращения: 06.09.2024).
8. Nguyen T.H., Pham C.K., Hoang T.T. A high-efficiency modular multiplication digital signal processing for lattice-based post-quantum cryptography. Cryptography. 2023, v. 7, no. 4, p. 46. DOI: http://dx.doi.org/10.3390/cryptography7040046.
9. Liang Z. et al. Number Theoretic Transform: Generalization, Optimization, Concrete Analysis and Applications. International Conference on Information Security and Cryptology. 2020, p. 415–432. DOI: http://dx.doi.org/10.1007/978-3-030-71852-7_28.
10. Basso A., Roy S. S. Optimized Polynomial Multiplier Architectures for Post-Quantum KEM Saber. 58th ACM/IEEE Design Automation Conference (DAC). 2021, p. 1285–1290. DOI: http://dx.doi.org/10.1109/DAC18074.2021.9586219.
11. Ishida T. et al. A Proposal of Solving NTRU Equation Using Matrix Representation Instead of Recursive Operation. 2023 Eleventh International Symposium on Computing and Networking Workshops (CANDARW). 2023, p. 227–231. DOI: http://dx.doi.org/10.1109/CANDARW60564.2023.00045.
12. Pornin T. Improved Key Pair Generation for Falcon, BAT and Hawk. Cryptology ePrint Archive. 2023. – 10 p. URL: https://eprint.iacr.org/2023/290 (дата обращения: 06.09.2024).

13. Satriawan A., Syafalni I., Mareta R., Anshori I., Shalannanda W. and Barra A. Conceptual Review on Number Theoretic Transform and Comprehensive Review on Its Implementations. IEEE Access, v. 11, p. 70288–70316, 2023. DOI: http://dx.doi.org/10.1109/ACCESS.2023.3294446.
14. Bisheh-Niasar M., Azarderakhsh R. and Mozaffari-Kermani M. High-Speed NTT-based Polynomial Multiplication Accelerator for Post-Quantum Cryptography. IEEE 28th Symposium on Computer Arithmetic (ARITH), Lyngby, Denmark. 2021, p. 94–101. DOI: http://dx.doi.org/10.1109/ARITH51176.2021.00028.
15. Li M., Tian J., Hu X., Cao Y. and Wang Z. High-Speed and Low-Complexity Modular Reduction Design for CRYSTALS-Kyber. IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Shenzhen, China. 2022, p. 1–5. DOI: http://dx.doi.org/10.1109/APCCAS55924.2022.10090253.

REFERENCES:

[1] Allende M. et al. Quantum-resistance in blockchain networks. Scientific Reports. 2023, v. 13, no. 1, p. 5664. DOI: http://dx.doi.org/10.1038/s41598-023-32701-6.
[2] Raavi M. et al. Security Comparisons and Performance Analyses of Post-Quantum Signature Algorithms. Lecture Notes in Computer Science. 2021, v. 12727, 24 p. DOI: http://dx.doi.org/10.1007/978-3-030-78375-4_17.
[3] Kirshanova E.A. et al. Post-quantum Signature Proposal for Standardization. Prikladnaya diskretnaya matematika. Prilozhenie. 2020, no. 13, p. 44–51. DOI: http://dx.doi.org/10.17223/2226308X/13/14 (in Russian).
[4] Ducas L. et al. Finding short integer solutions when the modulus is small. Annual International Cryptology Conference. 2023, p. 150–176. DOI: http://dx.doi.org/10.1007/978-3-031-38548-3_6.
[5] Alkim E. et al. Polynomial Multiplication in NTRU Prime: Comparison of Optimization Strategies on Cortex-M4. (2020). IACR Transactions on Cryptographic Hardware and Embedded Systems, 2021(1), p. 217–238. DOI: http://dx.doi.org/10.46586/tches.v2021.i1.217-238.
[6] Liang Z., Zhao Y. Number theoretic transform and its applications in lattice-based cryptosystems: A survey. arXiv preprint arXiv:2211.13546. 2022. – 35 p. DOI: http://dx.doi.org/10.48550/arXiv.2211.13546.
[7] Becker H. et al. Polynomial multiplication on embedded vector architectures. Cryptology ePrint Archive. 2021. – 29 p. URL: https://eprint.iacr.org/2021/998 (дата обращения: 06.09.2024).
[8] Nguyen T.H., Pham C.K., Hoang T.T. A high-efficiency modular multiplication digital signal processing for lattice-based post-quantum cryptography. Cryptography. 2023, v. 7, no. 4, p. 46. DOI: http://dx.doi.org/10.3390/cryptography7040046.
[9] Liang Z. et al. Number Theoretic Transform: Generalization, Optimization, Concrete Analysis and Applications. International Conference on Information Security and Cryptology. 2020, p. 415–432. DOI: http://dx.doi.org/10.1007/978-3-030-71852-7_28.
[10] Basso A., Roy S. S. Optimized Polynomial Multiplier Architectures for Post-Quantum KEM Saber. 58th ACM/IEEE Design Automation Conference (DAC). 2021, p. 1285–1290. DOI: http://dx.doi.org/10.1109/DAC18074.2021.9586219.
[11] Ishida T. et al. A Proposal of Solving NTRU Equation Using Matrix Representation Instead of Recursive Operation. 2023 Eleventh International Symposium on Computing and Networking Workshops (CANDARW). 2023, p. 227–231. DOI: http://dx.doi.org/10.1109/CANDARW60564.2023.00045.
[12] Pornin T. Improved Key Pair Generation for Falcon, BAT and Hawk. Cryptology ePrint Archive. 2023. – 10 p. URL: https://eprint.iacr.org/2023/290 (дата обращения: 06.09.2024).
[13] Satriawan A., Syafalni I., Mareta R., Anshori I., Shalannanda W. and Barra A. Conceptual Review on Number Theoretic Transform and Comprehensive Review on Its Implementations. IEEE Access, v. 11, p. 70288–70316, 2023. DOI: http://dx.doi.org/10.1109/ACCESS.2023.3294446.
[14] Bisheh-Niasar M., Azarderakhsh R. and Mozaffari-Kermani M. High-Speed NTT-based Polynomial Multiplication Accelerator for Post-Quantum Cryptography. IEEE 28th Symposium on Computer Arithmetic (ARITH), Lyngby, Denmark. 2021, p. 94–101. DOI: http://dx.doi.org/10.1109/ARITH51176.2021.00028.
[15] Li M., Tian J., Hu X., Cao Y. and Wang Z. High-Speed and Low-Complexity Modular Reduction Design for CRYSTALS-Kyber. IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Shenzhen, China. 2022, p. 1–5. DOI: http://dx.doi.org/10.1109/APCCAS55924.2022.10090253.

Поступила в редакцию – 14 сентября 2024 г. Окончательный вариант – 30 октября 2024 г.
Received – September 14, 2024. The final version – October 30, 2024.