УСКОРЕНИЕ АЛГОРИТМОВ ПРИВЕДЕНИЯ ЧИСЕЛ ПО МОДУЛЮ В ПОСТКВАНТОВОЙ СХЕМЕ ЭЦП FALCON

Финошин М. А.[1], Иванова И. Д.[2], Жуков И. Ю.[3]

DOI: 10.21681/2311-3456-2025-3-38-44

Цель исследования: *уменьшение объема предварительных вычислений и времени работы схемы подписи Falcon путем внедрения модифицированной версии алгоритма K-RED.*

Методы исследования: *оценка ресурсоемкости алгоритмов приведения чисел по модулю, математическое моделирование алгоритмов приведения, тестирование алгоритмов приведения в составе постквантовой схемы подписи.*

Результаты исследования: *умножение полиномов в факторкольце многочленов организовано в Falcon таким образом, что для его выполнения необходимо предварительно вычислить таблицы поиска, хранящие так называемые коэффициенты поворота. Алгоритмы приведения чисел по модулю, основанные на представлении чисел в специальной форме, требуют дополнительного масштабирования данных коэффициентов поворота на заданный фактор. На основе объема таблиц поиска, применяемых в процессе работы схемы подписи Falcon, в данном исследовании проведен сравнительный анализ ресурсоемкости алгоритмов Монтгомери и K-RED. Вследствие того, что расходы по памяти алгоритма K-RED превышают ресурсоемкость алгоритма Монтгомери почти в 2 раза, была рассмотрена его модификация, алгоритм K2-RED, которая позволяет добиться ускорения процесса приведения чисел по модулю при меньшем объеме масштабированных коэффициентов поворота. Доказана теорема, позволяющая обобщить алгоритм K-RED на случай, когда модуль приведения не является числом Прота. Также сформированы требования для размера факторов при представлении модуля приведения в форме модифицированного алгоритма K-RED, по которым было подобрано представление простых модулей в составе решения уравнения NTRU. Модифицированная версия алгоритма K-RED была реализована на языке программирования Си и внедрена в состав эталонной реализации схемы подписи Falcon. Проведено тестирование модифицированной версии алгоритма Falcon, в результате которого получено уменьшение времени выполнения процедур генерации ключей и проверки подписи.*

Научная новизна: *разработана модификация алгоритма приведения чисел по модулю K-RED, позволяющая применить модульную арифметику в форме K-RED к модулям общего вида. Данная модификация делает возможным внедрение быстрой арифметики в форме K-RED в процесс решения уравнения NTRU в составе схемы подписи Falcon, в ходе которого используются простые числа, не являющиеся числами Прота..*

Ключевые слова: *коэффициенты поворота, таблицы поиска, уравнение NTRU, преобразование NTT, алгоритм K-RED, алгоритм Монтгомери.*

Введение

Применение быстрых алгоритмов NTT, Кули-Тьюки и Джентельмена-Санде позволяет снизить исходную квадратичную сложность умножения полиномов до O(nlogn), однако достижение эффективного умножения исключительно посредством программной реализации по-прежнему остается сложной задачей. Так, последние исследования на тему ускорения операций умножения полиномов в схеме подписи Falcon направлены в основном на ее аппаратное усовершенствование за счет использования NEON-инструкций [1, 2] и RISC-V-инструкций [3]. Поскольку алгоритм Falcon является сильным кандидатом на внедрение в маломощные устройства [4], подход с использованием специализированных инструкций приводит к сужению области применения модифицированной версии Falcon. В частности, ее внедрение в иные устройства, не поддерживающие конкретные инструкции, может потребовать эмуляции NEON либо RISC-V. В таком случае предполагаемое ускорение вычислений может быть достигнуто не в полной мере.

Параллельно с этим в исследованиях производительности криптографических схем CRYSTALS при ускорении числового теоретического преобразования (от англ. Number Theoretic Transform, NTT) рассматривается алгоритм K-RED и его модификации в качестве алгоритма приведения чисел по модулю [5, 6]. Таким образом, на данный момент исследования в области ускорения схемы подписи Falcon охватывают только вопрос его аппаратной модификации, не учитывая возможности усовершенствования за счет внедрения новых математических подходов.

Настоящее исследование направлено на разработку и тестирование модифицированной версии схемы подписи Falcon, которая бы включала в себя

1 Финошин Михаил Александрович, старший преподаватель кафедры «Криптология и Кибербезопасность» Национального исследовательского ядерного университета «МИФИ», г. Москва, Россия. E-mail: mafinoshin@mephi.ru, ORCID 0000-0003-4374-1645.

2 Иванова Ирина Дмитриевна, ассистент кафедры «Высшая математика» Российского университета транспорта (МИИТ), г. Москва, Россия. E-mail: iid.ivanova@yandex.ru, ORCID 0000-0003-3022-8973.

3 Жуков Игорь Юрьевич, руководитель департамента разработок ООО Группа компаний «Инфотактика», г. Москва, Россия. E-mail: izhukov@infotaktika.ru, ORCID: 0000-0002-4429-8799.

более быстрый и менее ресурсоемкий алгоритм приведения чисел по модулю.

Ресурсоемкость алгоритмов приведения в составе схемы подписи Falcon

Алгоритм Falcon является финалистом конкурса NIST 2022 года среди постквантовых схем подписи, в котором также участвовали схемы подписи CRYSTALS-Dilithium и SPHINCS+. Алгоритм Falcon предлагает более компактные, нежели у других участников, размеры открытого ключа и подписей [7]. Алгоритм Falcon построен с использованием криптографии на основе теории решеток, и его безопасность обусловлена сложностью задачи нахождения кратчайшего целочисленного решения (от англ. Short integer problem, SIS). В задаче SIS вычисления проводятся над элементами кольца многочленов, и в алгоритме Falcon применяется факторкольцо $\mathbb{Z}_q[x] / (x^n + 1)$ по модулю простого числа q. При выполнении умножения полиномов в схеме подписи Falcon применяется быстрое преобразование Фурье (БПФ) и преобразование NTT.

БПФ используется при операциях с закрытым ключом (то есть в процедуре генерации подписи), а NTT – в операциях с открытым ключом (проверка подписи) и при генерации криптографических ключей. Преобразования Фурье работает с полем комплексных чисел C и вычисляется при помощи ω, примитивного корня степени $2n$ (в случае 1-го и 5-го уровней стойкости Falcon), определяемого как $\omega = e^{2i/2n}$ (где i – мнимая единица). Для многочлена a, представимого в виде вектора $a = (a[0], ..., a[n-1])$, форма преобразования Фурье $\hat{a}$ вычисляется как:

$$\hat{a}_i = \sum_{j=0}^{n-1} a_j (\omega^i)^j. \quad (1)$$

В случае преобразования NTT вычисления выполняются над конечным полем $\mathbb{Z}_q$, и ω определяется как примитивный корень степени $2n$ в $\mathbb{Z}_q$. Поскольку ω^i в преобразовании Фурье называют коэффициентами поворота (от англ. twiddle factor), в настоящем исследовании для обозначения степеней корня из единицы, применяемых в NTT, будет также использоваться данный термин.

Одним из базовых принципов оптимизации, применяемым во многих областях программирования, являются таблицы поиска (от англ. lookup table, LUT). Таблицы коэффициентов поворота для БПФ и NTT рассчитываются при помощи детерминированных функций и при ограниченном количестве входных значений, вследствие чего они также могут быть организованы как таблицы поиска [8]. В них ключами являются индекс частоты либо степень, в которую возводится корень из единицы $2n$-го порядка, а значения – коэффициентами поворота.

При нахождении формы NTT для коэффициентов полинома вычисления в формуле (1) выполняются по простому модулю q, который должен удовлетворять следующему условию [9]:

$$q \equiv 1 \bmod 2n. \quad (2)$$

В эталонной реализации[4] схемы подписи Falcon в качестве алгоритма приведения чисел по простому модулю используется алгоритм Монтгомери (с параметром $r = 2^k$, чтобы выполнялось условие НОД$(r,q) = 1$). В прошлой работе [10] было установлено, что в качестве более быстрой альтернативы может применяться алгоритм приведения чисел по модулю K-RED. Согласно исходной формулировке[5], алгоритм K-RED использует свойства чисел Прота, представимых как:

$$q = k \cdot 2^m + 1, \quad (3)$$

где $k < 2^m$ – малое нечетное натуральное число, m – натуральное число.

При применении оригинального алгоритма K-RED в ходе вычисление формы NTT для полинома необходимо одновременно применять две функции: K-RED и K-RED-2x, чтобы предотвратить возможное переполнение в случае большой длины полиномов n. При этом в функции K-RED вычисляется приведенное значение для kc:

$$kc \equiv kc_0 - c_1 \bmod q, \quad (4)$$

где $c_0 = c \bmod 2^m$, $c_1 = [\frac{c}{2^m}]$, а в функции K-RED-2x – для k^2 c:

$$k^2 \cdot c \equiv k^2 \cdot c_0 - kc_1 + c_2 \bmod q, \quad (5)$$

где $c_0 = c \bmod 2^m$, $c_1 = \frac{c}{2^m} \bmod 2^m$, $c_2 = [\frac{c}{2^{2m}}]$.

Таким образом, при прямом применении алгоритма K-RED выход является приведенной величиной не входного значения, а масштабированного – на фактор k либо k^2.

При быстрой реализации прямого преобразования NTT используется алгоритм Кули-Тьюки, который вычисляет коэффициенты полинома в форме NTT по формулам:

$$\hat{a}_i = \hat{a}_i' + \hat{a}_i'' \cdot \omega_i \bmod q, \quad (6)$$

$$\hat{a}_{i+\frac{n}{2}} = \hat{a}_i' - \hat{a}_i'' \cdot \omega_i \bmod q, \quad (7)$$

где $\hat{a}_i' = \sum_{j=0}^{\frac{n}{2}-1} a_{2j} (\omega^2)^{ij}$ и $\hat{a}_i'' = \sum_{j=0}^{\frac{n}{2}-1} a_{2j+1} (\omega^2)^{ij}$, а $i = 0, 1, ..., \frac{n}{2}-1$. Из формул (6) и (7) видно, что вычисление каждого нового коэффициента полинома в форме NTT требует одного умножения и операции сложения или вычитания. В обоих случаях, применяется ли алгоритм K-RED к промежуточному результату произведения либо к выходу операции сложения или вычитания, коэффициент $\hat{a}_i'$ будет требовать дополни-

4 Falcon source files (reference implementation). URL: https://falcon-sign.info/impl/vrfy.c.html

5 Longa P., Naehrig M. Speeding up the Number Theoretic Transform for Faster Ideal Lattice-Based Cryptography. DOI:10.1007/978-3-319-48965-0_8

тельного масштабирования на такой фактор k^s (где s – целое число), которым обладает произведение $\hat{a}_i'' \cdot \omega^i$.

В случае такого подхода возникают следующие проблемы:

- учет значения s на каждом этапе вычисления «бабочек» Кули-Тьюки и Джентельмена-Санде;
- затраты на промежуточное масштабирование в ходе работы прямого и обратного преобразований NTT;
- затраты на «выходное» масштабирование, после которого полином должен быть передан в общем виде обратно в процедуру генерации ключей.

Решить их возможно за счет масштабирования коэффициентов поворота в ходе их предварительного вычисления. Аналогично тому, как эталонная реализация схемы подписи Falcon содержит таблицы коэффициентов поворота, вычисленные в форме Монтгомери, так и в случае применения алгоритма K-RED вместо значений $\omega^i \bmod q$ необходимо использовать значения $k^s \cdot \omega^i \bmod q$ (где $s = -1$ либо $s = -2$ для функций K-RED и K-RED-2x соответственно).

Однако данное условие порождает проблему того, что использование алгоритма K-RED требует вычисления двух таблиц масштабированных коэффициентов поворота на каждое значение модуля фактор-кольца q. Поэтому, хотя алгоритм K-RED выполняется быстрее алгоритма Монтгомери [11], он требует в два раза большие расходы памяти.

Разработка модификации алгоритма K-RED

Среди различных предлагаемых исследователями модификаций алгоритма K-RED компромисс между объемом предварительно вычисляемых таблиц поиска и применением беззнаковой арифметики (которая внедрена в эталонную реализацию алгоритма Falcon) достигается в алгоритме K2-RED [12]. Функция, реализующая данный алгоритм, содержит два шага применения функции K-RED, так что выходом является приведенное значение: $\bar{\bar{c}} \equiv k^2 \cdot c \bmod q$. Поэтапно для параметров q, k, m и входного числа c в представлении (3) данный алгоритм реализуется следующим образом:

1. Вычисление c_0 и c_1: $c_0 = c \bmod 2^m$ и $c_1 = [\frac{c}{2^m}]$.
2. Первый шаг приведения по модулю
$$(\bar{c} \equiv kc \bmod q)\colon \bar{c} = kc_0 - c_1.$$
3. Вычисление c_0' и c_1': $c_0' = \bar{c} \bmod 2^m$ и $c_1' = [\frac{\bar{c}}{2^m}]$.
4. Второй шаг приведения по модулю
$$(\bar{\bar{c}} \equiv k^2 \cdot c \bmod q)\colon \bar{\bar{c}} = kc_0' - c_1'.$$
5. Алгоритм возвращает с $\bar{\bar{c}}$.

В отличие от классического K-RED, в котором во избежание переполнения на отдельных итерациях преобразования NTT необходимо заменять функцию K-RED на функцию K-RED-2x, алгоритм K2-RED не требует дополнительных шагов приведения чисел по модулю. Это позволяет сократить объем предварительно вычисляемых таблиц масштабированных коэффициентов до одной таблицы поиска на каждый модуль приведения. Кроме того, применение алгоритма K2-RED облегчает задачу анализа и устранения дополнительных факторов k^s, которые «зашумляют» выходное значение вследствие дополнительных итераций использования K-RED-2x. Таким образом, в ходе масштабирования на фактор n^{-1} внутри обратного преобразования NTT:

- дополнительное масштабирование на степень фактора k не требуется, если алгоритм K2-RED применялся исключительно в ходе прямого и обратного преобразований NTT;
- дополнительное масштабирование на степень фактора k производится в соответствии с применением алгоритма K2-RED в ходе покомпонентного умножения полиномов.

Преобразование NTT применяется в алгоритме Falcon в процессе генерации ключей и проверки подписи. При проверке подписи используется фиксированный модуль приведения, имеющий вид $q = 12289 = 3 * 2^{12} + 1$. Таким образом, факторы k и m для него также являются фиксированными, к тому же k является малым нечетным числом. В то же время процедура генерации ключей применяет хотя и заранее определенное, но целое множество простых модулей.

Закрытый ключ задается как кортеж полиномов (f, g, F, G), где полиномы f и g генерируются случайным образом как элементы факторкольца многочленов $\mathbb{Z}[x] / (x^n + 1)$ (при этом f должен быть обратимым многочленом в кольце $\mathbb{Z}_q[x] / (x^n + 1)$, а полиномы F и G вычисляются путем решения уравнения NTRU [13]:

$$fG - gF = q \bmod (x^n + 1). \quad (8)$$

Для решения уравнения применяется китайская теорема об остатках [14], вследствие чего выбирается подмножество простых чисел $\{p_i\}$, по модулю которых проводятся операции над полиномами, в частности, их умножение. Именно на этом этапе в модуле keygen.c, реализующем процедуру генерации ключей, применяется преобразование NTT. Это возможно за счет вида применяемых простых модулей:

$$p_i \equiv 1 \bmod 2n. \quad (9)$$

Поскольку размерность пространства n является степенью двойки (512 для версии Falcon 1-го уровня стойкости и 1024 для 5-го уровня стойкости), тождество (9) удовлетворяет условию (3) для применения

алгоритма приведения числа по модулю K-RED, а значит и K2-RED. Однако ввиду того, что модули $\{p_i\}$ подбираются в алгоритме Falcon таким образом, чтобы быть немногим меньше 2^{31}, подавляющее большинство из них не являются числами Прота, вследствие чего фактор k в представлении (3) оказывается большим нечетным числом. Данный факт затрудняет применение как алгоритма K-RED, так и алгоритма K2-RED при решении уравнения NTRU.

В исходной статье, предлагающей алгоритм приведения K-RED, упоминается, что данный алгоритм может быть обобщен также на случай, когда модуль приведения представим в виде:

$$q = k \cdot 2^m \pm l, \quad (10)$$

где k и l натуральные нечетные и $k \geq 3$, а $l \geq 1$.

С этой целью докажем следующую теорему:

Теорема. (Модульная арифметика для модуля вида $q = k \cdot 2^m \pm l$). Для модуля q вида (10) и любого целого числа с выполняется:

$$kc \equiv kc_0 \mp lc_1 \bmod q, \quad (11)$$

где $c_0 = c \bmod 2^m$, $c_1 = [\frac{c}{2^m}]$.

Доказательство. Очевидно, что выполняется тождество:

$$c_1 \, q \equiv 0 \bmod q. \quad (12)$$

Используем вид представления модуля (10) и произведем подстановку:

$$c_1 \cdot (k \cdot 2^m + l) \equiv 0 \bmod q. \quad (13)$$

Разнесем слагаемые по разные стороны тождества:

$$c_1 \cdot k \cdot 2^m \equiv -lc_1 \bmod q. \quad (14)$$

Воспользуемся определением переменной c_1:

$$\frac{c - c_0}{2^m} \cdot k \cdot 2^m \equiv (c - c_0) \cdot k \equiv -lc_1 \bmod q. \quad (15)$$

Таким образом, получим:

$$kc \equiv kc_0 - lc_1 \bmod q. \quad (16)$$

Полученный вывод свидетельствует о том, что в общем случае, чтобы осуществить приведение по модулю по алгоритму K-RED либо K2-RED, необходимо предварительно определить параметры k, m и l в представлении модуля приведения по формуле (10). При решении уравнения NTRU данные значения разумно хранить в таблицах поиска.

Тестирование модифицированного алгоритма K-RED

При модификации алгоритма K-RED были совмещены техники с двойным приведением по модулю (как в алгоритме K2-RED) и представлением модуля приведения по формуле (10) в соответствии с доказанной теоремой. Как уже было упомянуто, простые модули для решения уравнения NTRU подобраны таким образом, чтобы быть немногим меньше числа 2^{31}. В процессе анализа массива PRIMES модуля keygen.c рассматривались представления p_i в виде формулы (10) для различных значений m – длины младшей части исходного числа c_0 из представления (4). К сожалению, данные простые числа не являются числами Прота, потому подобрать такое значение m, чтобы $l = \pm 1$, не представляется возможным. Однако, как было доказано в прошлом разделе данной работы, алгоритм K-RED приведения числа по модулю может быть обобщен на любое l. При условии, что умножение на k и l в формуле (16) должно быть эффективным, k и l следует подбирать таким образом, чтобы они были в некотором смысле «малы» относительно применяемого модуля p. Таким образом, если в представлении простого числа порядка 2^{31} фактор k имеет порядок 2^7, то в этом случае отношение k к p будет меньше, чем в предлагаемом разложении модуля q = 12289 с k = 3.

В ходе анализа обнаружено, что начиная с m = 24 приведенные в массиве PRIMES простые числа могут быть разложены при помощи одинаковых факторов: например, k = 127 при m = 24. При увеличении m опытным путем было получено, что с уменьшением фактора k происходит рост процентного отношения l к p. Таким образом, у разложения с m = 24 были обнаружены следующие преимущества:

1) равенство факторов k для всех p из массива PRIMES при фиксированном m = 24 (это позволяет сократить расходы на таблицы поиска, содержащие параметры k, k^{-2}, k^{-4}, m для различных p);
2) простота умножения на фактор k = 127:

$$127 \times k = k \times 2^7 - k;$$

3) относительно небольшой размер l (при больших значениях m значение l в процентном отношении к p возрастает).

Поскольку подобранное представление простых чисел p_i позволяет использовать одно и то же значение m для всех модулей приведения, для облегчения вычислений при выделении остатка от деления на 2^m возможно применять операцию логического «И» с предварительно вычисленной маской. Ниже приведен фрагмент реализации модифицированного алгоритма K-RED на языке программирования Си, используемом в эталонной реализации схемы подписи Falcon:

```
z0 = k * (z & mask) - l * (z >> m).
z1 = k * (z0 & mask) - l * (z0 >> m).
return z1.
```

В данном фрагменте z – исходное приводимое по модулю значение (масштабировано на фактор k^{-2} за счет умножения на масштабированный коэффициент поворота), k, m, l – факторы из представления

(10), mask – маска, вычисляемая по значению фактора m, **z1** – выходное значение, являющееся приведением по модулю q величины $z \cdot k^{-2}$.

Тестирование предлагаемой модификации алгоритма K-RED проводилось при помощи функции «test_nist_KAT» из модуля «test_falcon.c» эталонной реализации Falcon. Данная функция использует тестовые векторы согласно представленным NIST рекомендациям. Эти рекомендации были представлены в ходе проведения соревнования между постквантовыми криптографическими схемами [15]. Тестирование выполнено для векторов длиной n = 1024, алгоритм K-RED был внедрен в прямое и обратное преобразования NTT. Результаты вычислительного эксперимента приведены для процессора Intel® Celeron(R) N4000 CPU @ 1.10GHz × 2.

В таблице 1 представлены результаты тестирования эталонной реализации с использованием алгоритма Монтгомери при приведении чисел по модулю в процессе вычисления формы NTT и выполненной реализации с использованием модифицированного алгоритма K-RED. Единица измерения времени выполнения – такты процессора.

Частично данное уменьшение времени выполнения процедур генерации ключей и проверки подписи было достигнуто за счет сохранения дополнительных предвычисленных значений. Среди них в модуле keygen.c: фиксированные факторы k и m (применяются в процессе применения алгоритма приведения K-RED), по одной таблице поиска на k^{-2} и k^{-4} (нужны при масштабировании коэффициентов поворота) и одна таблица поиска со значениями фактора l для элементов массива PRIMES. В модуле vrfy.c масштабированная таблица коэффициентов поворота предварительно вычисляется и хранится аналогично таблице поиска для коэффициентов в форме Монтгомери. При этом значение фактора l фиксировано (равно 1), потому в сравнении с эталонной реализацией алгоритма Falcon изменения в затратах памяти не происходят.

В таблице 2 приведен сравнительный анализ затрат по памяти и по времени схемы подписи Falcon с использованием модифицированного алгоритма приведения K-RED в сравнении с эталонной реализацией.

Как видно из таблицы 2, форма модуля простого числа (10) незначительно влияет на прирост производительности в выполнения преобразования NTT, что позволяет утверждать, что предлагаемая модификация алгоритма K-RED с неединичным фактором l может быть эффективной альтернативой применяемому в эталонной реализации алгоритму Монтгомери. При этом общий прирост производительности в модуле keygen.c меньше, чем в vrfy.c, за счет более сложного алгоритма, также включающего в себя вычисления с плавающей запятой с применением БПФ.

Таблица 1.

Результаты тестирования эталонной реализации схемы подписи Falcon и с применением модифицированного алгоритма K-RED

Модуль, содержащий вычисления в форме NTT		**Эталонная реализация + Монтгомери**	**Эталонная реализация + модифицированный K-RED**
vrfy.c	Прямое преобразование NTT	6 845 т.	5 921 т.
	Обратное преобразование NTT	3 641 т.	3 271 т.
	Всего	17 630 т.	16 431 т.
keygen.c	Прямое преобразование NTT	11 452 т.	10 082 т.
	Обратное преобразование NTT	8 149 т.	7 415 т.
	Всего	56 588 т.	54 167 т.

Таблица 2.

Сравнительный прирост производительности эталонной реализации с применением модифицированного алгоритма K-RED

	Модуль, содержащий вычисления в форме NTT					
	vrfy.c			keygen.c		
Изменение затрат по памяти	–			+3 LUT		
Изменение времени выполнения	NTT	INTT	Всего	NTT	INTT	Всего
	14 %↓	10 %↓	7 %↓	12 %↓	9 %↓	4 %↓

Выводы

В результате настоящего исследования предложен и протестирован в составе схемы подписи Falcon модифицированный алгоритм K-RED. В ходе работы проведен анализ ресурсоемкости классического алгоритма K-RED в сравнении с алгоритмом Монтгомери, применяемым в эталонной реализации Falcon. Рассмотрен алгоритм K2-RED, являющийся модификацией алгоритма K-RED и позволяющий уменьшить объем требуемых предварительно вычисляемых значений, которые применяются при работе преобразования NTT. Доказана теорема, позволяющая обобщить алгоритм K-RED на простые числа, не являющиеся числами Прота. Выполнена модификация алгоритма K-RED, объединившая в себе техники из алгоритма K2-RED и выводы из доказанной в настоящем исследовании теоремы.

Модифицированная версия алгоритма K-RED была внедрена в состав эталонной реализации схемы подписи Falcon и протестирована, в результате чего получено уменьшение времени выполнения как процедуры проверки подписи, так и процедуры генерации ключей. Дополнительные расходы по памяти при этом составили 3 таблицы поиска, хранящие факторы формы K-RED. В дальнейших исследованиях планируется использовать модифицированную в настоящей работе схему подписи Falcon для сравнения производительности протокола TLS в случае применения классических и постквантовых схем подписи.

Литература

1. Accelerating Falcon on ARMv8 / Y. Kim, J. Song, S.C. Seo // IEEE Access. 2022. Vol. 10. p. 44446–44460. DOI: 10.1109/ACCESS.2022.3169784.
2. Nguyen D.T., Gaj K. Fast Falcon Signature Generation and Verification Using ARMv8 NEON Instructions // International Conference on Cryptology in Africa. 2023. P. 417–441. DOI: 10.1007/978-3-031-37679-5_18.
3. Wang L.N. et al. Support Post Quantum Cryptography with SIMD Everywhere on RISC-V Architectures // Workshop Proceedings of the 53rd International Conference on Parallel Processing. 2024. P. 23–32. DOI: 10.1145/3677333.3678149.
4. Security Comparisons and Performance Analyses of Post-Quantum Signature Algorithms / M. Raavi, S. Wuthier, P. Chandramouli [et al] // International Conference on Applied Cryptography and Network Security. 2021. Vol. 12727. p. 424–447. DOI: 10.1007/978-3-030-78375-4_17.
5. HyperNTT: A Fast and Accurate NTT/INTT Accelerator with Multi-Level Pipelining and an Improved K2-RED Module / D.N. Nguyen, H.L. Pham, V.T.D. Le [et al] // 2024 International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC). 2024. P. 1–6. DOI: 10.1109/ITC-CSCC62988.2024.10628429.
6. High-Speed and Low-Complexity Modular Reduction Design for CRYSTALS-Kyber / M. Li, J. Tian, X. Hu [et al] // 2022 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS). 2022. P. 1–5. DOI: 10.1109/APCCAS55924.2022.10090253.
7. FalconSign: An Efficient and High-Throughput Hardware Architecture for Falcon Signature Generation / Y. Ouyang, Y. Zhu, W. Zhu [et al] // IACR Transactions on Cryptographic Hardware and Embedded Systems. 2024. Vol. 2025, № 1. P. 203–226. DOI: 10.46586/tches.v2025.i1.203-226.
8. Land G., Sasdrich P., Güneysu T. A Hard Crystal – Implementing Dilithium on Reconfigurable Hardware // International Conference on Smart Card Research and Advanced Applications. 2022. P. 210–230. DOI: 10.1007/978-3-030-97348-3_12.
9. 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: 10.48550/arXiv.2211.13546.
10. Иваненко В.Г., Иванова И.Д., Иванова Н.Д. Вычисления над полиномами в постквантовых схемах подписи // Вопросы кибербезопасности. 2024. № 4(62) С. 65–70. DOI: 10.21681/2311-3456-2024-4-65-70.
11. 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. Vol. 7. No. 4. p. 46. DOI: 10.3390/cryptography7040046.
12. Bisheh-Niasar M., Azarderakhsh R., Mozaffari-Kermani M. High-Speed NTT-based Polynomial Multiplication Accelerator for Post-Quantum Cryptography // 2021 IEEE 28th symposium on computer arithmetic (ARITH). 2021. P. 94–101. DOI: 10.1109/ARITH51176.2021.00028.
13. Teixeira C., Gazzoni Filho D.L., Hernandez J.C.L. Improving FALCON's Key Generation on ARMv8-A Platforms // Anais do XXIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. 2023. P. 528–533. DOI: 10.5753/sbseg.2023.233093.
14. Efficient Hardware RNS Decomposition for Post-Quantum Signature Scheme Falcon / S. Coulon, P. He, T. Bao [et al] // 2023 57th Asilomar Conference on Signals, Systems, and Computers. 2023. P. 19–26. DOI: 10.1109/IEEECONF59524.2023.10476845.
15. PQC-HA: A Framework for Prototyping and In-Hardware Evaluation of Post-Quantum Cryptography Hardware Accelerators / R. Sattel, C. Spang, C. Heinz [et al] // arXiv preprint arXiv:2308.06621. 2023. 20 p. DOI: 10.48550/arXiv.2308.06621.

ACCELERATING MODULAR REDUCTION FOR FALCON SIGNATURE SCHEME

Finoshin M.A.[6], Ivanova I.D.[7], Zhukov I.Y.[8]

6 Mihail A. Finoshin, senior lecturer of the Cryptology and Cybersecurity Department at NRNU MEPhI, Moscow, Russia. E-mail: MAFinoshin@mephi.ru, ORCID 0000-0003-4374-1645.

7 Irina D. Ivanova, assistant of Department of Higher Mathematics, Russian University of Transport (MIIT), Moscow, Russia. E-mail: iid.ivanova@yandex.ru, ORCID 0000-0003-3022-8973

8 Igor Yu. Zhukov, head of the Development Department, group of companies «Infotaktika», Moscow, Russia. E-mail: izhukov@infotaktika.ru, ORCID: 0000-0002-4429-8799.

Keywords: *twiddle factors, lookup tables, NTRU equation, NTT, K-RED modular reduction, Montgomery multiplication.*

Purpose of the study: *precomputation reducing and execution time speeding up of Falcon signature scheme by implementing a modified version of the K-RED algorithm.*

Methods of research: *resource intensity evaluation of modular reduction algorithms, mathematical modeling of modular reduction algorithms, testing of modular reduction algorithms as part of the post-quantum signature scheme.*

Results: *multiplication of polynomials in the polynomial quotient ring is organized in Falcon in such a way that its execution requires precomputed lookup tables that store so-called twiddle factors. Modular reduction algorithms based on representing numbers in a special form require additional scaling of these twiddle factors by a given factor. Based on the size of the lookup tables used in Falcon signature scheme, a comparative analysis of the resource intensity of the Montgomery and K-RED algorithms has been conducted. Due to the fact that the memory consumption of the K-RED algorithm is almost twice that of the Montgomery algorithm, the K2-RED algorithm which allows for faster modular reduction with a smaller volume of scaled twiddle factors has been considered. A theorem that generalizes the K-RED algorithm to the case where the reduction modulus is not a Proth number has been proven. Additionally, requirements for the size of modified K-RED factors have been established, based on which representations of prime moduli in the NTRU equation solution have been selected. The modified K-RED algorithm has been implemented in C language and integrated into the Falcon reference implementation. Testing of modified Falcon has been conducted, resulting in a reduction in the execution time of key generation and signature verification procedures.*

Scientific novelty: *a modified version of the K-RED algorithm that allows the application of modular arithmetic in the K-RED form to general modules has been developed. The developed version of K-RED algorithm makes it possible to use fast arithmetic in the K-RED form during the process of solving the NTRU equation as part of Falcon.*

References

1. Kim, Y., Song, J., & Seo, S. C. (2022). Accelerating Falcon on ARMv8. IEEE Access, 10, 44446-44460. DOI: 10.1109/ACCESS.2022.3169784.
2. Nguyen, D.T., & Gaj, K. (2023, July). Fast Falcon Signature Generation and Verification Using ARMv8 NEON Instructions. In International Conference on Cryptology in Africa, 417-441. DOI: 10.1007/978-3-031-37679-5_18.
3. Wang, L.N., Li, J.H., Kuan, C.B., & Su, Y.C. (2024, August). Support Post Quantum Cryptography with SIMD Everywhere on RISC-V Architectures. In Workshop Proceedings of the 53rd International Conference on Parallel Processing, 23-32. DOI: 10.1145/3677333.3678149.
4. Raavi, M., Wuthier, S., Chandramouli, P., Balytskyi, Y., Zhou, X., & Chang, S.Y. (2021, June). Security Comparisons and Performance Analyses of Post-Quantum Signature Algorithms. In International Conference on Applied Cryptography and Network Security, 424–447. DOI: 10.1007/978-3-030-78375-4_17.
5. Nguyen, D.N., Pham, H.L., Le, V.T.D., Lam, D.K., Tran, T.H., & Nakashima, Y. (2024, July). HyperNTT: A Fast and Accurate NTT/INTT Accelerator with Multi-Level Pipelining and an Improved K2-RED Module. In 2024 International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC), 1-6. DOI: 10.1109/ITC-CSCC62988.2024.10628429.
6. Li, M., Tian, J., Hu, X., Cao, Y., & Wang, Z. (2022, November). High-Speed and Low-Complexity Modular Reduction Design for CRYSTALS-Kyber. In 2022 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 1–5. DOI: 10.1109/APCCAS55924.2022.10090253.
7. Ouyang, Y., Zhu, Y., Zhu, W., Yang, B., Zhang, Z., Wang, H., & Liu, L. (2025). FalconSign: An Efficient and High-Throughput Hardware Architecture for Falcon Signature Generation. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2025(1), 203–226. DOI: 10.46586/tches.v2025.i1.203-226.
8. Land, G., Sasdrich, P., & Guneysu, T. (2021, November). A Hard Crystal – Implementing Dilithium on Reconfigurable Hardware. In International Conference on Smart Card Research and Advanced Applications, 210–230. DOI: 10.1007/978-3-030-97348-3_12.
9. Liang, Z., & Zhao, Y. (2022). Number Theoretic Transform and Its Applications in Lattice-based Cryptosystems: A Survey. arXiv preprint arXiv:2211.13546. DOI: 10.48550/arXiv.2211.13546.
10. Ivanenko, V.G., Ivanova, I.D., & Ivanova N.D. (2024). Optimization of Computations over Polynomials in Post-Quantum Signature Scheme. Voprosy kiberbezopasnosti, (4), 62, 65–70. DOI: 10.21681/2311-3456-2024-4-65-70.
11. Nguyen, T.H., Pham, C.K., & Hoang, T.T. (2023). A High-Efficiency Modular Multiplication Digital Signal Processing for Lattice-Based Post-Quantum Cryptography. Cryptography, 7(4), 46. DOI: 10.3390/cryptography7040046.
12. Bisheh-Niasar, M., Azarderakhsh, R., & Mozaffari-Kermani, M. (2021, June). High-Speed NTT-based Polynomial Multiplication Accelerator for Post-Quantum Cryptography. In 2021 IEEE 28th symposium on computer arithmetic (ARITH), 94-101. DOI: 10.1109/ARITH51176.2021.00028.
13. Teixeira, C., Gazzoni Filho, D.L., & Hernandez, J.C.L. (2023, September). Improving FALCON's Key Generation on ARMv8-A Platforms. In Anais do XXIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, 528-533. DOI: 10.5753/sbseg.2023.233093.
14. Coulon, S., He, P., Bao, T., & Xie, J. (2023, October). Efficient Hardware RNS Decomposition for Post-Quantum Signature Scheme Falcon. In 2023 57th Asilomar Conference on Signals, Systems, and Computers, 19-26. DOI: 10.1109/IEEECONF59524.2023.10476845.
15. Sattel, R., Spang, C., Heinz, C., & Koch, A. (2023). PQC-HA: A Framework for Prototyping and In-Hardware Evaluation of Post-Quantum Cryptography Hardware Accelerators. arXiv preprint arXiv:2308.06621. DOI: 10.48550/arXiv.2308.06621.