DOI: 10.18372/2410-7840.18.11095
УДК 519.6:004.056.55

МЕТОДЫ ИЗВЛЕЧЕНИЯ КОРНЯ С ОСТАТКОМ ИЗ МНОГОРОЗРЯДНЫХ ЧИСЕЛ ДЛЯ РЕШЕНИЯ ЗАДАЧ АССИМЕТРИЧНОЙ КРИПТОГРАФИИ

Степан Винничук, Александр Корнейко, Евгений Максименко

На данный момент асимптотически самыми быстрыми методами факторизации многоразрядных последовательностей являются методы, построенные на фундаментальных соотношениях алгоритма Ферма. Одной из наиболее сложных операций в методе факторизации Ферма является процедура извлечения квадратного корня. Существующие методы вычисления корней относятся к числу итерационных и требуют выполнения достаточно большого количества трудоемких арифметических операций умножения и деления, что в свою очередь существенно влияет на их временную оценку. Одним из способов повышения производительности существующих методов извлечения квадратных корней может быть использование процедуры прямого вычисления корня, не использующей операции умножения или деления больших чисел. Предложен новый метод вычисления квадратного корня без использования операций умножения и деления больших чисел, являющийся модификацией метода извлечения корня «в столбик». Проведен сравнительный анализ описанного модифицированного метода «в столбик» с существующим диагональным методом (методом Терещенко). Предлагается использование данного метода в процедурах анализа ассиметричных криптоалгоритмов.

Ключевые слова: *факторизация, ассиметричная криптография, квадратный корень, деление с остатком, диагональный метод, метод «в столбик», модифицированный метод, многоразрядные числа.*

Вступление. Научная задача построения эффективного вычислительного алгоритма извлечения квадратного корня из больших многоразрядных чисел достаточно важна для проблем современной асимметричной криптографии.

Ведь, например, при проведении криптоанализа наиболее эффективным на данный момент вычислительным методом факторизации общего решета числового поля (General Number Field Sieve, GNFS) криптомодулей длиной сотни бит многих современных асимметричных криптографических алгоримов шифрования и цифровой подписи, стойкость которых основана на сложности факторизации (RSA, ESIGN, GQ1 і GQ2, GPS1, GPS2 и др.), задача извлечения корня становится чрезвычайно сложной и трудоемкой задачей, сравнимой по затратам ресурсов со сложностью задачи просеивания чисел [1-3]. Кроме того, сложность выполнения задачи извлечения корня характерна и для универсального метода факторизации Ферма [1, 2].

При выполнении вычислительной задачи дискретного логарифмирования, на сложности которой основана стойкость, например, асимметричных криптоалгоритмов Диффи-Хеллмана (Diffie-Hellman), Эль-Гамаля (Elgamal), Мэсси-Омуры (Massey-Omura) и др., также существует сопутствующая задача эффективного нахождения квадратного корня [3, 4].

А стойкость криптоалгоритмов Рабина (Rabin), RW (Rabin-Williams) и Фиата-Шамира (Fiat-Shamir) вообще основывается на сложности задачи поиска квадратных корней по модулю составного числа [4].

В научной литературе [5] для этих целей рекомендуется использовать итерационный алгоритм

$$x_{n+1} = \left(x_n + \frac{A}{x_n} \right) / 2 \ (n = 0,1,...), \qquad (1)$$

где A – число, из которого извлекают корень.

Если же стоит задача определения целого числа *X* такого, что

$$\begin{cases} X^2 \le A \\ (X+1)^2 \ge A \end{cases}, \qquad (2)$$

то после достижения требуемой точности в ходе итерационного процесса (1) из полученного значения корня выделяют целую часть $[x]$ и определяют один из вариантов остатков

$$\begin{cases} r = A - [X]^2 \\ r = ([X]+1)^2 - A \end{cases}. \qquad (3)$$

Кроме указанного итерационного метода определения корня с остатком из многоразрядного числа существуют также ряд способов целочисленного извлечения корня.

Поэтому проанализируем известные методы извлечения квадратного корня из многоразрядных чисел и оценим их эффективность.

Диагональный метод Терещенко (алгоритм Д). В большинстве языков программирования в основе процедуры извлечения квадратного корня лежит итерационный алгоритм Ньютона и его модификации [5, 6]. В случае алгебраических операций с большими числами в современных реализациях библиотек математических функций применяется алгоритм Карацубы («Karatsuba

Square Root»), использующий процедуры Быстрого Преобразования Фурье и являющийся модификацией того же итерационного метода Ньютона [7, 8].

В работе [9] А.Н. Терещенко был предложен метод быстрого вычисления квадрата/куба числа, существенно отличающегося от указанных выше. В основе предложенного метода лежит алгоритм диагонального возведения в квадрат, который эффективно используется для вычисления обратной задачи – вычисления квадратного и кубического корней. Отличительной особенностью данного метода является отказ от операций умножения и деления, что является особенно актуальным при работе с большими числами.

Для описания диагонального метода вычисления квадратного корня (алгоритма Д) введем несколько дополнительных обозначений:

– X_i, Y_i – битовые последовательности размерностью i+1 и 2(i+1);

– x_i, y_i – биты в битовых последовательностях X и Y;

– R_i – промежуточная битовая последовательность размерностью 2(i+1).

Тогда алгоритм Д можно представить в виде следующей последовательности шагов:

1. Исходная битовая последовательность Y_i, представляется в двоичной системе исчисления и разбивается на пары бит Y_i = $= y_0\ y_1 \mid y_2\ y_3 \mid y_4\ y_5 \mid y_6\ y_7 \mid \ldots$ начиная со старших разрядов числа;

2. $X_0 = x_0 = 1$, $R_0 = y_0\ y_1$, $R_0 = R_0 - X_0$;

3. $i = i + 1$, $R_i = R_{i-1} * 4 + y_{2i}\ y_{2i+1}$, $dY_i = X_{i-1} * 4 + 1$;

4. Если $R_i \geq dY_i$, то $R_i = R_i - dY_i$, $x_i = 1$, иначе $x_i = 0$;

5. $X_i = X_{i-1} * 2 + x_i$. Если биты $y_{2i}\ y_{2i+1}$ не последние, то перейти к 3 шагу.

По окончании алгоритма переменная X_i будет содержать целую часть от квадратного корня, а R_i – остаток.

Ниже приведен пример вычисления квадратного корня из числа $123_{10} = 0111\ 1011_2 = (11_{10})^2 + 2_{10} = (1011_2)^2 + 10_2$.

1. $Y_i = y_0\ y_1 \mid y_2\ y_3 \mid y_4\ y_5 \mid y_6\ y_7 \mid \ldots$ = 01 | 11 | 10 | 11.

2. $x_0 = 1$; $X_0 = 1$; $R_0 = y_0\ y_1 = 01$; $dY_0 = 1$; $R_0 = R_0 - y_0\ y_1 = 01 - 1 = 0$.

Таблица 1

Вычисление квадратного корня из числа N=123

i	$y_{2i}\ y_{2i+1}$ $R_i = R_{i-1} * 4 + y_{2i}\ y_{2i+1}$	$dY_i = X_{i-1} * 4 + 1$ $R_i = R_i - dY_i$ при $R_i \geq dY_i$ x_i	$X_i = X_{i-1} * 2 + x_i$
i=0	$y_0\ y_1 = 0\ 1$ **$R_0 = 0$**	**$x_0 = 1$**	**$X_0 = 1$**
i=1	$y_2\ y_3 = 1\ 1$ **$R_1 = R_0 * 4 + y_2\ y_3 = 11$**	$dY_1 = X_0 * 4 + 1 = 101$ $11 < 101$, **$x_1 = 0$**	**$X_1 = X_0 * 2 + x_1 = 10$**
i=2	$y_4\ y_5 = 1\ 0$ $R_2 = R_1 * 4 + y_4\ y_5 = 1110$	$dY_2 = X_1 * 4 + 1 = 1001$ $1110 > 1001$, **$x_2 = 1$** **$R_2 = R_2 - dY_2 = 101$**	**$X_2 = X_1 * 2 + x_2 = 101$**
i=3	$y_6\ y_7 = 1\ 1$ $R_3 = R_2 * 4 + y_6\ y_7 = 10111$	$dY_3 = X_2 * 4 + 1 = 10101$ $10111 > 10101$, **$x_3 = 1$** **$R_3 = R_3 - dY_3 = 10$**	**$X_3 = X_2 * 2 + x_3 = 1011$**

В результате мы получили следующие значения: $123_{10} = 1111011_2 = X_3^2 + R_3$, где корень $X_3 = 11_{10} = 1011_2$, а остаток $R_3 = R_3 - dY_3 = 2_{10} = 10_2$.

Метод извлечения корня «в столбик» (алгоритм С). Пусть *b* – некоторое основание системы счисления.

Тогда любое натуральное число можно представить в виде разложения в ряд

$$\sum_{k=0}^{n} a_k b^k, \qquad (4)$$

где a_k $(k = 0 \div n)$ – коэффициенты разложения по основанию b.

Если N представить в виде

$$N = (xb + y)^2 = N = x^2b^2 + 2bxy + y^2,$$

то $N - x^2b^2 = y \cdot (2bx + y)$. (5)

На основании соотношения (5) можно найти целое значение корня с остатком из произвольного целого числа, которое удовлетворяет условиям (2).

Тогда алгоритм С описывается следующей последовательностью шагов:

1. Число $\sum_{k=0}^{n} a_k b^k$ представить в виде

$$\sum_{k=0}^{n} a_k b^k = \sum_{k=0}^{m} (a_{2k} + a_{2k+1} b) b^{2k}, \qquad (6)$$

где m=[(n-1)/2], причем при четных n $a_{2m+1} = 0$. Разбить его на блоки по два коэффициента разложения (6) (k – й блок содержит коэффициенты a_{2k} и a_{2k+1}).

2. Для числа, равного $a_{2m} + a_{2m+1} b$ найти такое x, что $x^2 \le a_{2m} + a_{2m+1} b$ и $(x+1)^2 \ge a_{2m} + a_{2m+1} b$.

3. i=m-1, $A = a_{2m} + a_{2m+1} b - x^2$.

4. Вычислить $Z = A \cdot b^2 + a_{2i} + a_{2i+1} b$.

5. Определить y такое, что

$$\begin{cases} Z \ge y \cdot (2bx + y) \\ Z \le (y+1) \cdot (2bx + y + 1) \end{cases}. \qquad (7)$$

6. Определить $A = Z - y \cdot (2bx + y)$. (8)

7. Определить $x = x \cdot b + y$. (9)

8. i=i-1.

9. Если $i \ge 0$ перейти у шагу 4, а иначе корнем будет число x, а остатком число A.

Рассмотрим два примера определения корня с остатком из числа N методом «в столбик» для случая основания системы счисления 10.

Пример 1. Пусть N=12345678. Число N содержит четное число цифр n=8, а m=3. В соответствии с п. 1 алгоритма С это число N можно представить в виде $12345678 = 12 \cdot 100^3 + 34 \cdot 100^2 + 56 \cdot 100^1 + 78 \cdot 100^0$, что определяет блоки по два коэффициента разложения: 12, 34, 56 и 78. Согласно п. 2 алгоритма для числа 12 определяем x=3 ($x^2 \le 12$ и $(x+1)^2 \ge 12$). Согласно п.3 определяем i=3-1=2; $A=12-3^2=3$. Результаты расчетов для п.4-9 алгоритма С представим в таблице 2, где при каждом из значений i определяются числа Z, y, x, A согласно соотношений (7) – (9).

Таблица 2

Вычисление квадратного корня с остатком методом в столбик для N = 12345678

i	$Z = A \cdot b^2 + a_{2i} + a_{2i+1} b$	2xb	y = [Z/(2xb+y)]	A = Z-y*(2bx+y)	x = x*b + y
2	3·100+34 =334	2·3·10=60	5	334-325=9	35
1	9·100+56 =956	2·35·10=700	1	956-701=255	351
0	255·100+78 =25578	2·351·10=7020	3	25578-21069 = **4509**	**3513**

Следовательно, $12345678 = 3513^2 + 4509 = 12341169 + 4509$, т.е. целочисленное значение корня x=3513, а остаток A=4509.

Пример 2. Пусть N = 123456789. Число N содержит нечетное число цифр n=9 и m=4. В соответствии с п.1 алгоритма С его можно представить в виде $123456789 = 1 \cdot 100^4 + 23 \cdot 100^3 + 45 \cdot 100^2 + 67 \cdot 100^1 + 89 \cdot 100^0$, что определяет блоки по два коэффициента разложения: 1, 23, 45, 67 и 89. Согласно п.2 алгоритма С для числа 1 определяем x=1 ($x^2 \le 1$ и $(x+1)^2 \ge 1$).

Согласно п.3 алгоритма определяем i=4-1=3; $A=1-1^2=0$. Результаты расчетов для п.4-8 алгоритма С представим в таблице 3, где при каждом из значений i определяются числа Z, y, x, A.

Таблица 3

Последовательность определения корня и остатка для N = 123456789

i	$Z = A \cdot b^2 + a_{2i} + a_{2i+1} b$	2xb	y = [Z/(2xb+y)]	A = Z-y*(2bx+y)	x = x*b + y
3	0·100+23 =23	2·1·10=20	1	23-21=2	11
2	2·100+45 =245	2·11·10=220	1	245-221=24	111
1	24·100+67 =2467	2·111·10=2220	1	2467-2221=246	1111
0	246·100+89 = 24689	2·1111·10 = 22220	1	24689-22221 = **2468**	**11111**

Следовательно, $123456789 = 11111^2 + 2468 = 123454321 + 2468$, т.е. целочисленное значение корня x=11111, а остаток A=2468.

Аналогичные вычисления «в столбик» для примеров 1 и 2 представлены на рис. 1а и 1б соответственно.

а)

$\sqrt{12\,34\,56\,78}$	x	y	$2bx+y$
-9	3	-	-
$\times 65 \mid -334$ $5 \mid 325$	35	5	65
$\times 701 \mid -956$ $1 \mid 701$	351	1	701
$\times 7023 \mid -25578$ $3 \mid 21069$	3513	3	7023

$N - x^2 = 4509 \quad x=3513$

б)

$\sqrt{1\,23\,45\,67\,89}$	x	y	$2bx+y$
-1	1	-	-
$\times 21 \mid -23$ $1 \mid 21$	11	1	21
$\times 221 \mid -245$ $1 \mid 221$	111	1	221
$\times 2221 \mid -2467$ $1 \mid 2221$	1111	1	2221
$\times 22221 \mid -24689$ $1 \mid 22221$	11111	1	22221

$N - x^2 = 2468 \quad x=11111$

Рис.1. Графическое представление алгоритма определения корня с остатком для чисел, содержащих четное (а) и нечетное (б) количество коэффициентов в разложении вида (4)

Оценим временную сложность метода «в столбик» на основании определения числа для операций умножения, вычитания и деления больших чисел, в случае поразрядных вычислений, где под разрядом понимается коэффициент в разложении (4).

Для операций вычитания определим наихудший вариант. Максимальное число разрядов вычитаемых чисел при i = m равно двум. После первого вычитания максимально возможное число разрядов не может превысить 4. На всех последующих шагах число разрядов не может превысить 3+m-i. Поэтому число операций поразрядного вычитания без учета разрядных переносов оценивается сверху величиной 2+4+5+…+(2+m--1)+(2+m)= 2+(m-1)(m+6)/2.

Операции умножения выполняются в соотношениях (8), (9) и косвенно в (7).

В соотношении (8) имеет место умножение разряда на большое число, содержащее число разрядов, равное m-i на шаге i. На всех шагах число разрядных умножений (без учета разрядных переносов) равно m(m-1)/2.

В соотношении (9) имеет место разрядный сдвиг и одно сложение.

Наиболее сложной среди арифметических операций является деление многоразрядных чисел. Но при определении y, удовлетворяющего соотношениям (7), делитель является числом, зависимым от y. Поэтому результат можно получить только в итерациях, число которых редко может превысить 2.

С учетом того, что в результате деления получается значение, которое не должно превысить основание b, процедуру деления целесообразно выполнять путем сравнения делимого числа с числами, кратными делителю, где коэффициент кратности выбирается методом деления пополам.

Тогда вычислительная сложность операции деления будет пропорциональна $m\log_2 b$. В результате суммирования операций поразрядного умножения при вычислениях значений кратных делителю значений получим их число, равное $\log 2b \cdot m(m-1)/2$.

Общее число операций в тактах компьютера будет величиной $O(m^2)$, где коэффициент пропорциональности превысит число $\log_2 b$ в несколько раз.

Отдельного рассмотрения требует случай b=2. Тогда на шаге определения y достаточно одного сравнения делимого и делителя, что собственно и реализовано в методе, представленном в работе [9]. Но в таком случае явно большим окажется значение m для числа, для которого определяется корень и остаток.

Поэтому эффективность метода, представленного в [9] и метода «в столбик» при разном числе разрядов требует дополнительного исследования.

Модифицированный метод извлечения корня «в столбик» (алгоритм М). Для рассмотренного выше метода определения «в столбик» целочисленного корня из натурального числа наиболее трудоемкой вычислительной операцией является шаг 5 алгоритма С при определении y, удовлетворяющего соотношениям (7).

Поэтому предлагается метод, позволяющий ускорить именно этот этап вычисления при расчетах y, что и определяет сущность модификации алгоритма С.

Пусть как и для метода определения целочисленного корня из натурального числа «в столбик» число представлено рядом (4) разложения по основанию b. Оно представлено блоками, содержащими по два коэффициента разложения вида (6), где число блоков m=[(n-1)/2] + 1.

Тогда алгоритм М описывается такой последовательностью шагов:

1. Для чисел $R1 = a_{2m-2} + b\cdot(a_{2m-1} + b\cdot(a_{2m} + a_{2m+1}b))$ и $R2 = R1+1$ с использованием встроенной математической функции sqrt(*) определить: $x_{m-1} = \lfloor\sqrt{R1}\rfloor$; $r_1 = 2\sqrt{R1}$; $r_2 = 2\sqrt{R2}$; $A_{m-1} = R1 - x_{m-1}^2$.

2. i=m-1.

3. Вычислить

$$Z_i = A_i \cdot b^2 + a_{2i} + a_{2i+1}b = \sum_{k=2i}^{m+i} a_k^* b^{k-2i},$$

где при k=2i и k=2i+1 $a_k^* = a_k$, а при больших k a_k^* являются коэффициентами числа A_i.

4. Определить y, соответствующее условиям (7) по правилам:

4.1. Вычислить $R1_i = a^*_{m+i-2} + b\cdot(a^*_{m+i-1} + b\cdot(a^*_{m+i} + b\cdot a^*_{m+i+1}))$ и $R2_i = R1_i + 1$;

4.2. Найти $y1_i = \lfloor R1_i / r2\rfloor$ и $y2_i = \lfloor R2_i / r1\rfloor$;

4.3. Принять $y_i = y2_i$ и определить значение $R_i = y2_i \cdot (2bx_i + y2_i)$;

4.4. Если $y1_i = y2_i$ перейти к п.М5, а иначе к п. 4.5.

4.5. Сравнить R_i и Z_i. Если $R_i > Z_i$ принять $y_i = y1_i$ и определить значение $R_i = y1_i \cdot (2bx_i + y1_i)$. Перейти к п. 5.

5. Определить $A_{i-1} = Z_i - R_i$;

6. Определить $x_{i-1} = x_i \cdot b + y_i$;

7. i = i - 1.

8. Если $i > 0$ перейти к п. 3, а иначе корнем будет число x_0, а остатком число A_0.

Рассмотрим два примера определения корня с остатком из числа N модифицированным методом «в столбик» для случая основания системы счисления 10.

Пример 3. Пусть N =120999999. Число N содержит нечетное число цифр n=9, т.е. m=4. В соответствии с п. 1 алгоритма М число можно представить в виде $120999999 = 1\cdot100^4 + 20\cdot100^3 + 99\cdot100^2 + 99\cdot100^1 + 99\cdot100^0$, что определяет блоки по два коэффициента разложения: 01, 20, 99, 99 и 99.

Согласно п. 1 алгоритма М определяем: R1=120, R2=121; $x_{m-1} = \lfloor\sqrt{R1}\rfloor$=10.0; $r_1 = 2\sqrt{R1}$ ≈21.9; $r_2 = 2\sqrt{R2}$ =22,0; $A_{m-1} = R1 - x^2$ =20.

Согласно п. 2 алгоритма М определяем i=4-1=3. Результаты расчетов для пп. 3-8 алгоритма М представлены в таблице 4, где при каждом из значений i определяются числа Z, R1, y, x, A, а также выделены разряды: числа R1 – жирным курсивом, числа Z, которые не используются в R1 – подкрашены цветом, а разряды решения – прямым жирным.

Таблица 4

Последовательность определения корня и остатка для *N* = 120999999

		Разряды N										
		8	7	6	5	4	3	2	1	0		
a_i		1	2	0	9	9	9	9	9	9		
R1		1	2	0							$r_1 = 2\sqrt{R1} \approx 21.9$; $r_2 = 2\sqrt{R2}$ =22	x_3=10
$A_3 = R1 - x_3^2$			2	0								
i=3	Z_3 ($R1_3$)		***2***	***0***	***9***	9					$y1_3=[R1_3/r_2]$=9 $y2_3=[R2_3/r_1]$=9	y_3=9	x_2 = 109
	$R_3 = y_3(2bx_3+y_3)$		1	8	8	1							
	A_2			2	1	8							
i=2	$Z_2(R1_2)$			***2***	***1***	***8***	9	9			$y1_2=[R1_2/r_2]$= 9 $y2_2=[R2_2/r_1]$=10	y_2=9	x_1 = 1099
	Сравнение 21899 и (2180+10)*10 = 21900: 21900>21899												
	$R_2=y_2(2bx_2+y_2)$			1	9	7	0	1					
	A_1				2	1	9	8					
i=1	$Z_1(R1_1)$				***2***	***1***	***9***	8	9	9	$y1_1=[R1_1/r_2]$=9 $y2_2=[R2_1/r_1]$=10	y_1=9	**x_0 = 10999**
	Сравнение 219899 и (21980+10)*10 = 219900: 219900>219899												
	$R_1=y_1(2bx_1+y_1)$				1	9	7	9	0	1			
	A_0					**2**	**1**	**9**	**9**	**8**			

Следовательно, $120999999 = 10999^2 + 21998$, т.е. целочисленное значение корня x=10999, а остаток A=21998. Пример 3 подбирался с условием, что в нем будет максимально возможное число сравнений, предусмотренное в п. 4.5 алгоритма М.

В большинстве же случаев условия для сравнений возникают редко, что подтверждается в примере 4.

Пример 4. Пусть N=1209999999. Число N содержит нечетное число цифр n=10, а m = 4. В

соответствии с п.1 алгоритма его можно представить в виде $1209999999 = 12\cdot100^4 + 09\cdot100^3 + 99\cdot100^2 + 99\cdot100^1 + 99\cdot100^0$, что определяет блоки по два коэффициента разложения: 12, 09, 99, 99 и 99.

Согласно п. 1 алгоритма М для числа 100 определяем: R1=1209, R2=1210; $x_{m-1} = \left[\sqrt{R1}\right] = 34.0$; $r_1 = 2\sqrt{R1} \approx 69.54$; $r_2 = 2\sqrt{R2} \approx 69{,}57$ и $A = R1 - x^2 = 53$.

Отметим, что x=34 удовлетворяет требованиям $x^2 \le 1209$ и $(x+1)^2 \ge 1209$.

Согласно п. 2 алгоритма М определяем i=4-1=3. Результаты расчетов для пп. 3-8 алгоритма М представлены в таблице 5, где при каждом из значений i определяются соответствующие значения чисел Z, R1, y1, y2, y, x, A, а также (аналогично как и в табл. 4) выделены разряды: числа R1 – жирным курсивом, числа Z, которые не используются в R1 – подкрашены цветом, а разряды решения – прямым жирным.

Таблица 5

Последовательность определения корня и остатка для *N* = 1209999999

		Разряды N												
		9	8	7	6	5	4	3	2	1	0			
a_i		1	2	0	9	9	9	9	9	9	9			
R1		1	2	0	9							$r_1 = 2\sqrt{R1} \approx 69.54$; $r_2 = 2\sqrt{R2} \approx 69{,}57$		$x_3 = 34$
$A_3 = R1 - x_3^2$				5	3									
i=3	$Z_3(R1_3)$			***5***	***3***	***9***	9					$y1_3 = [R1_3/r_2] = 7$		
	$R_3 = y_3(2bx_3 + y_3)$			4	8	0	9					$y2_3 = [R2_3/r_1] = 7$	$y_3 = 7$	$x_2 = 347$
	A_2				5	9	0							
i=2	$Z_2(R1_2)$				***5***	***9***	***0***	9	9			$y1_2 = [R1_2/r_2] = 8$		
	$R_2 = y_2(2bx_2 + y_2)$				5	5	5	8	4			$y2_2 = [R2_2/r_1] = 8$	$y_2 = 8$	$x_1 = 3478$
	A_1					3	5	1	5					
i=1	$Z_1(R1_1)$					***3***	***5***	***1***	5	9	9	$y1_1 = [R1_1/r_2] = 5$		
	$R_1 = y_1(2bx_1 + y_1)$					3	4	7	8	2	5	$y2_1 = [R2_1/r_1] = 5$	$y_1 = 5$	**x_0= 34785**
	A_0							**3**	**7**	**7**	**4**			

Следовательно, $1209999999 = 34785^2 + 3774$, т.е. целочисленное значение корня x = 34785, а остаток A = 3774.

Хотя найденные целочисленные значения квадратного корня и остатка из чисел в примерах 3 и 4 являются правильными, правило определения y в алгоритме М требует обоснования. Опишем структуру x, Z и R1, а именно число разрядов в них и правило их формирования в зависимости от значения i.

На шаге М1 при m>0 у найденного значения x_{m-1} всегда 2 разряда. При произвольных $i \ge 0$ на шагах пп. 3-5 алгоритма М количество разрядов x_i равно m-i и увеличивается на единицу на шаге п. 6. Следовательно, на шагах пп. 3-5 алгоритма М при произвольном i корень x_i можно представить в виде

$$x_i = x_{m-1}b^{m-i-1} + \sum_{k=i}^{m-2} c_k b^{k-i}, \qquad (10)$$

где $c_k < b\ (k = i \div m)$.

Тогда

$$x_{m-1}b^{m-i-1} \le x_i < (x_{m-1}+1)b^{m-i-1}. \qquad (11)$$

Полагая также

$$Z_i = \sum_{k=2i-2}^{m+i} a_k^* b^{k-2i+2} =$$

$$= a_{m+i}^* b^{m-i+2} + a_{m+i-1}^* b^{m-i+1} + a_{m+i-2}^* b^{m-i} +$$

$$+ \sum_{k=2i-2}^{m+i-3} a_k^* b^{k-2i+2} = R1b^{m-i} + \sum_{k=2i-2}^{m+i-3} a_k^* b^{k-2i+2},$$

где $a_k^* < b\ (k = 2i \div m+i)$, получим оценки для Z_i

$$R1b^{m-i} \le Z_i < (R1+1)b^{m-i}. \qquad (12)$$

Пусть $N_i = \sum_{k=2i}^{n} a_k b^{k-2i}$

и

$$\sqrt{N_i} = bx_i + y_i + s_i, \qquad (13)$$

где s_i – действительное положительное число, меньшее за единицу. Тогда

$$\sqrt{Rb^{2m-2i}} = b^{m-i}\sqrt{R} \le \sqrt{N_i} \le$$

$$\le \sqrt{(R+1)b^{2m-2i}} = b^{m-i}\sqrt{R+1},$$

т.е. $b^{m-i}r_1 \le 2\sqrt{N_i} \le b^{m-i}r_2$. (14)

Исходя из соотношения (13) можно получить представление Z_i через x_i, y_i, s_i

$$Z_i = N_i - x_i^2 b^2 = 2x_i b(y_i + s_i) + (y_i + s_i)^2 =$$
$$= (y_i + s_i)(2x_i b + (y_i + s_i)) \; (i = 0 \div m-1),$$

используя которое, а также оценки (3.9) и (3.11) получаем нижнюю оценку для y_i

$$(y_i + s_i) = \frac{Z_i}{2bx_i + y_i + s_i} \ge \frac{Z_i}{2(bx_i + y_i + s_i)} =$$
$$= \frac{Z_i}{2\sqrt{N_i}} \ge \frac{R1_i}{r2}.$$

Но $[y_i + s_i] = y_i$. Поэтому $y_i \ge \left[\frac{R1_i}{r2}\right]$.

Известно, что при определении квадратного корня из некоторого числа A итерационный процесс $x_{k+1} = (x_k + A/x_k)/2 \; (k \ge 0)$ характеризуется квадратичной скоростью сходимости, а при $k > 0$ $x_k \ge x_{k+1} \ge \sqrt{A}$.

Следовательно,

$$(y_i + s_i) = \left(\sqrt{N_i} + \frac{N_i}{\sqrt{N_i}}\right)/2 - bx_i \le$$
$$\le \left(bx_i + \frac{N_i}{bx_i}\right)/2 - bx_i = \frac{N_i - (bx_i)^2}{2bx_i} =$$
$$= \frac{Z_i}{2bx_i} \le \frac{R2_i b^{m-i}}{2bx_i}.$$

Но $2bx_i \ge r_1 \cdot b^{m-i}$ при m-i > 2 . Поэтому

$$(y_i + s_i) \le \frac{R2_i b^{m-i}}{2bx_i} \le \frac{R2_i b^{m-i}}{r_1 \cdot b^{m-i}} = \frac{R2_i}{r_1} \; (m - i > 2)$$

и

$$y_i = [y_i + s_i] \le [R2_i / r_1] \; (m - i > 2). \qquad (15)$$

Достоверность оценки (15) для $m\text{-}i \le 2$ проверялась с помощью численных экспериментов для b=4÷12.

Целью численных экспериментов было также определение относительного числа случаев, когда $[R1_i / r_2] < [R2_i / r_1]$ и проверка условия $y_i \le [R2_i / r_1]$.

В первой серии численных экспериментов анализировались все числа N в диапазоне от b^6 до b^8–1 при b=4÷12, для которых подсчитывалось общее число случаев, когда $[R1_2 / r_2] < [R2_2 / r_1]$, а также максимальное число таких случаев для фиксированного $R1_2$, которое оказалось равным b^2. Определялись y_i, R, r_1, r_2, $R1_2, R2_2$, $y1_2 = [R1_2 / r_2]$, $y2_2 = [R2_2 / r_1]$, $R1_1, R2_1$, $y1_1 = [R1_1 / r_2]$, $y2_1 = [R2_1 / r_1]$, а также относительная величина числа случаев, когда $y_2 = y1_2$ и $y_1 = y1_1$.

Было установлено, что относительная величина количества вариантов значений чисел N, для которых $[R1_i / r_2] < [R2_i / r_1]$ для i=2 практически совпадает с аналогичной величиной для случая i = 1 и близка к значению $\left(b^8 - b^6\right)/b^2$. Она равна общему количеству вариантов чисел N, деленному на квадрат основания b.

Поэтому была проведена еще одна серия численных экспериментов для b=10÷35, где анализировались все числа N в диапазоне от b^4 до b^6–1. Для количества вариантов значений чисел N, для которых $y1 < y2$ получено подтверждение того, оно близко к отношению общего количества анализируемых вариантов чисел N, деленного на квадрат основания b.

Показано также, что для произвольных i: $y_i \le [R2_i / r_1]$, а относительная величина числа случаев, когда $y_i = y2_i$, больше, чем когда $y_i = y1_i$, что позволило обосновать выбор варианта y_i на шаге 4.3 алгоритма М.

Анализ числа выполняемых операций алгоритмами М, С и Д. Сравним количество выполняемых операций в алгоритмах М и С определения целочисленного корня из натурального числа «в столбик». Алгоритм М содержит ряд дополнительных операций, которые отсутствуют в алгоритме С. Это определение R, квадратного корня из него и из R+1, значений r_1 и r_2. Но все эти операции выполняются однократно и в формуле для вычислительной сложности представляются константами.

Дополнительные операции в алгоритме М, число которых растет с увеличением количества разрядов N, это операции шагов п. 4 по определению $R1_i, R2_i, y1_i$. Для произвольного i их определение требует не более 3 операций сложения, 4 операций умножения, 2 делений и определения целой части частного. Следовательно, число таких операций для произвольного N пропорционально числу его разрядов при разложении по некоторому основанию b. При этом значение y_i определяется сразу из условия $y1_i = y2_i$ для относительного числа вариантов R_i, примерно равного

$(b^2-1)/b^2$. В остальных случаях может потребоваться сравнение $R_i = y2_i \cdot (2bx_i + y2_i)$ с Z_i, а при $R_i > Z_i$ вычисление величины $R_i = y1_i \cdot (2bx_i + y1_i)$. Т.е. и в таких относительно редких случаях число выполняемых операций для алгоритма метода М будет меньшим, чем для алгоритма С уже при b≥4. При этом алгоритм М будет выполняться более эффективно по сравнению с алгоритмом С при больших основаниях b.

При сравнении количества операций алгоритма Д и алгоритма М можно однозначно утверждать, что при основании системы счисления b=2 диагональный метод будет эффективнее. Но в при $b=2^{10}$ число пар разрядов в модифицированный методе «в столбик» уменьшается в десять раз, в связи с чем он может оказаться эффективнее.

Для получения сравнительных оценок временной сложности методов были проведены вычислительные эксперименты, результаты которых анализируются ниже.

Численное сравнение временной сложности модифицированного метода определения корня с остатком «в столбик» и диагонального метода. В численных экспериментах рассматривались числа, содержащие от 80 до 1060 двоичных разрядов. При разработке приложения, реализующего алгоритм Д, на шаге п. 2 на каждой итерации выполнялось не более 7 операций присвоения. На шаге п. 5 – 2 операции присвоения. Наиболее трудоемким был шаг п. 4, где при $x_i = 1$ выполнялась операция вычитания $R_i=R_i–dY_i$. При этом в программном коде на языке С использовались только операции сравнения и присвоения.

Численные эксперименты реализованы и проведены на одном и том же компьютере с характеристиками: процессор Intel(R) Core i3-3110M CPU 2.40 GHz, ОЗУ 6 ГБ (доступно 2.41 ГБ), 32-разрядная ОС. С учетом того, что время расчета одного варианта задания оказалось меньшим 0.001 сек, одна и той же задача решалась 10^4 раз. Для диапазона чисел от 2^{500} до 2^{1000} установлены зависимости времени расчета от числа разрядов. В случае диагонального метода $T(n)=O(n^{k1})$, где k1=2.079÷2.092. Аналогичные зависимости для модифицированного метода «в столбик» $T(n)=O(n^{k2})$, где k2=1.79÷1.82. В табл. 6 представлены фактические данные о времени расчета 10^4 повторений расчетов квадратного корня с остатком для чисел с количеством разрядов, кратных 200, где Т1 – время расчета для диагонального метода (алгоритма Д), а Т2 – для модифицированного метода «в столбик» (алгоритма М).

Таблица 6

Сравнительные данные о времени расчета

Число разрядов n	200	400	600	800	1000
Время расчета T_1, с	0,234	1,438	3,547	6,36	10,251
Время расчета T_2, с	0,02	0,066	0,131	0,22	0,341
T_1 / T_2	11,527	21,887	27,035	28,87	30,097

Выводы. Полученные данные численных экспериментов (табл. 6) позволяют утверждать, что предложенный модифицированный метод «в столбик» определения квадратного корня с остатком из многоразрядных чисел является наиболее эффективным среди рассмотренных в статье.

Теоретические предпосылки высокой эффективности этого метода основаны на эффективности алгоритма, представленного соотношением (1), которое можно преобразовать к виду

$$x_{n+1} = \left(x_n + \frac{A}{x_n} \right) / 2 = x_n + \frac{A - x_n^2}{2x_n} \, (n = 0,1,...) .$$

При этом в модифицированном методе «в столбик» вместо прибавляемой к x_n части $\frac{A-x_n^2}{2x_n}$ используется только часть разрядов числа $A-x_n^2$, которые обеспечивают вычисление точных значений разрядов в итерационном значении x_{n+1}, что избавляет от необходимости деления большого A на x_n.

ЛИТЕРАТУРА

[1]. Корнейко А.В. Анализ известных вычислительных методов факторизации многоразрядных чисел / А.В. Корнейко, А.В. Жилин // Моделювання та інформаційні технології: Збірник наукових праць. – Вип. 61. – К.: ІПМЕ, 2011. – С. 3-13

[2]. Ишмухаметов Ш.Т. Методы факторизации натуральных чисел: учебное пособие / Ш.Т. Ишмухаметов. – Казань: Казан. ун-т, 2011. – 190 с.

[3]. Нестеренко А.Ю. Теоретико-числовые методы в криптографии: учебное пособие / А.Ю. Нестеренко. – М.: МГИЭМ, 2012. – 224 с.

[4]. Горбенко І.Д. Прикладна криптологія : монографія / І.Д. Горбенко, Ю.І. Горбенко. Видання 2-ге. – Харків: Форт, 2012. – 878 с.

[5]. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. – 2-е. – М.: Вильямс, 2005. – 1296 с.
[6]. Дональд Э. Кнут Искусство программирования. Том 2. – М.: Мир, 1979. – 727 с.
[7]. Square Root algorithm for C. [Электронный ресурс]. – Режим доступа: http://www.codeproject. Com / Articles / 570700 / SquareplusRootplusalgorithmplusforplusC.
[8]. Best Square Root Method. Algorithm. Function. [Электронный ресурс]. – Режим доступа: http://www.codeproject.com/Articles/69941/Best-Square-Root-Method-Algorithm-Function-Precisi.
[9]. Терещенко А.Н. Быстрое вычисление квадратного и кубического корней без использования операций умножения и деления // Искусственный интеллект. – 2006. – Вып. 3. – С. 670-680.

REFERENCES

[1]. Korneiko A. V., Zhilin A. V. (2011), Analysis of the known methods for computing the factorization of large numbers, Collection of scientific works Institute of Modelling Problems inPower Engineering, Vol. 61, pp. 3-13.
[2]. Ishmukhametov Sh. T. (2011), Methods of factoring integers: tutorial, Kazan, Kazan, 190 p.
[3]. Nesterenko A. Yu. (2012), Theoretical and numerical methods in cryptography: tutorial, MGIEM, Moskow, 224 p.
[4]. Horbenko I.D., Horbenko Yu. I. (2012), Prykladna kryptolohiia: monohrafia. Edition 2, Fort, Kharkiv, 878 p.
[5]. Thomas Cormen, Charles. Leiserson, Ronald L. Rivest (2005), Introduction to Algorithms is a book by Edition 2, Viliams, Moscow, 1296 p.
[6]. Knut Donald E. (1997) Art of Computer Programming. Vol. 2, Mir, Moskow, 727 p.
[7]. Square Root algorithm for C. [Electronic resource]. — Access to resourceshttp: //www.codeproject. com / Articles / 570700 / SquareplusRootplusalgorithmplusforplusC.
[8]. Best Square Root Method. Algorithm. Function. [Electronic resource]. — Access to resources http://www.codeproject.com/Articles/69941/Best-Square-Root-Method-Algorithm-Function-Precisi.
[9]. Tereschenko A.N. (2005), Bystroe vychislenie kvadratnogo i kubicheskogo kornei bez ispolzovaniia operatcii umnozheniia i deleniia [Fast Calculation of Square and Cube Roots Without Multiplication and Division], Iskusstvennyi intellekt, No. 3, pp. 670-680.

МЕТОДИ ОБЧИСЛЕННЯ КОРЕНЯ ІЗ ЗАЛИШКОМ З БАГАТОРОЗРЯДНИХ ЧИСЕЛ ДЛЯ РІШЕННЯ ЗАДАЧ АСИМЕТРИЧНОЇ КРИПТОГРАФІЇ

На даний момент асимптотично найшвидшими методами факторизації багаторозрядних послідовностей є методи, побудовані на фундаментальних співвідношеннях алгоритму Ферма. Однією з найбільш складних операцій в методі факторизації Ферма є процедура обчислення квадратного кореня. Існуючі методи обчислення коренів відносяться до числа ітераційних і вимагають виконання досить великої кількості трудомістких арифметичних операцій множення і ділення, що в свою чергу суттєво впливає на їх тимчасову оцінку. Одним із способів підвищення продуктивності існуючих методів обчислення квадратних коренів може бути використання процедури прямого визначення кореня, що не використовує операції множення або ділення великих чисел. Запропоновано новий метод обчислення квадратного кореня без використання операцій множення і ділення великих чисел, що є модифікацією методу вилучення кореня «в стовпчик». Проведено порівняльний аналіз описаного модифікованого методу «в стовпчик» з існуючим діагональним методом (методом Терещенко). Пропонується використання даного методу в процедурах аналізу асиметричних криптоалгоритмів.

Ключові слова: факторизація, асиметрична криптографія, квадратний корінь, ділення з залишком, діагональний метод, метод «в стовпчик», модифікований метод, багаторозрядні числа.

METHODS OF EXTRACTING ROOT WITH THE RESIDUES FROM MULTI-BIT NUMBERS TO MEET THE CHALLENGES OF ASYMMETRIC CRYPTOGRAPHY

Currently asymptotically the fastest factorization methods of multi-bit sequences are methods based on the fundamental ratios of Fermat algorithm. One of the most complex operations in the factorization method of Fermat is a procedure of extracting the square root. Existing methods of extracting the roots are refer to the number of iterative and require the implementation the quite large number the most complex arithmetic operations of multiplication and division. This in turn significantly affects on their temporal assessment. One of the ways of improving the productivity of existing methods for extracting the square root would be using a direct calculation of root without using operations of multiplication or division large numbers. Offered a new method for calculating the square root without using multiplication and division large numbers, which is a modification of the method of extracting roots "in the column." Made a comparative analysis of the described modified method "in a column" with the existing diagonal method (method of Tereshchenko).
Offered to use this method in the procedures for analysis of asymmetric cryptographic algorithms.

Keywords: factorization, asymmetric cryptography, square root, division with remainder, diagonal method, method "in the column", modified method, multi-bit numbers.

Винничук Степан Дмитрович, доктор технічних наук, с.н.с., завідувач відділу Інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України.
E-mail: vynnychuk@i.ua.
Винничук Степан Дмитриевич, доктор технических наук, с.н.с., заведующий отделом Института проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины.
Vinnichuk Stepan, Dr. Science in Eng., Head of Science Department of The G. Pukhov Institute for Energy Modeling Engineering of The National Academy Sciences of Ukraine.

Корнейко Олександр Васильович, кандидат технічних наук, професор, в.о. вченого секретаря Інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України.
E-mail: alex_korneiko@meta.ua.
Корнейко Александр Васильевич, кандидат технических наук, профессор, и.о. ученого секретаря Института проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины.
Korneiko Oleksandr, Ph.D. in Eng., Professor, Scientific Secretary of The G. Pukhov Institute for Energy Modeling Engineering of The National Academy Sciences of Ukraine.

Максименко Євген Васильович, аспірант Інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України.
E-mail: maksimenco@gmail.com.
Максименко Евгений Васильевич, аспирант Института проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины.
Maksymenko Yevgen, Doctoral Student of The G. Pukhov Institute for Energy Modeling Engineering of The National Academy Sciences of Ukraine.