УДК 519.688
DOI: 10.24160/1993-6982-2018-5-79-88

Об оценках сложности алгоритмов извлечения квадратных корней в конечных полях и кольцах вычетов

С.Б. Гашков, А.Б. Фролов, Е.П. Попова

Представлен обзор известных алгоритмов извлечения квадратных корней из квадратичных вычетов по простому модулю и модулю степени простого числа, используемых посредством китайской теоремы об остатках для вычисления квадратных корней по любому числовому модулю. К ним относятся вероятностные алгоритмы Тонелли и Чиполлы и детерминированные алгоритмы для вычисления квадратных корней из квадратичных вычетов в кольцах вычетов по простому модулю, не сравнимому с единицей по модулю 8, а также строящиеся на их основе алгоритмы извлечения квадратного корня по модулю, равному степени простого числа.
Приведены новые алгоритмы извлечения квадратных корней в полях характеристики, сравнимой с тройкой по модулю 4, и квадратных корней из вычетов, сравнимых с единицей по модулю 8, в кольцах вычетов по модулю степени двойки. Для вероятностных алгоритмов даны оценки сложности и вероятности успешного выполнения при первой попытке.
Рассмотрена задача извлечения квадратных корней, встречающаяся в теоретико-числовых алгоритмах криптографии, используемых в сочетании с китайской теоремой об остатках для извлечения квадратного корня в кольцах вычетов по произвольному модулю.
Показаны известные вероятностные и детерминированные алгоритмы извлечения квадратных корней по модулю простого числа и оценки их сложности и вероятности успешного завершения с первой попытки. Обоснованы новые находящие применение в эллиптической криптографии алгоритмы извлечения квадратных корней в полях характеристики, сравнимой с 3 по модулю 4, с оценками сложности. Изучены алгоритмы извлечения квадратных корней в полях четного порядка также с оценками сложности. Продемонстрированы алгоритмы извлечения квадратных корней из квадратичных вычетов в кольцах по модулю степени нечетного простого числа и по модулю степени двойки.

Ключевые слова: конечное поле, кольцо вычетов, извлечение квадратного корня, квадратный корень по модулю простого числа, квадратный корень по модулю степени простого числа, подъем решений, оценка сложности алгоритма.

Для цитирования: Гашков С.Б., Фролов А.Б., Попова Е.П. Об оценках сложности алгоритмов извлечения квадратных корней в конечных полях и кольцах вычетов // Вестник МЭИ. 2018. № 5. С. 79—88. DOI: 10.24160/1993-6982-2018-5-79-88.

About the Complexity of Square Root Extraction Algorithms in Finite Fields and Residue Rings

S.B. Gashkov, A.B. Frolov, E.P. Popova

The article presents a review of the known algorithms for extracting square roots of the quadratic residues modulo primes and prime powers. According to the Chinese remainder theorem, such algorithms can be constructed to calculate square roots modulo any integer. These include the Tonelli and Chipolla probabilistic algorithms and deterministic algorithms for calculating the square roots of quadratic residues in residue rings modulo a prime number not congruent to 1 modulo 8, as well as the algorithms for extracting a square root modulo a power of a prime number constructed on their basis.
The article presents new algorithms for extracting square roots in finite fields of characteristic congruent to 3 modulo 4, for square roots of residues congruent to 1 modulo 8, and for square roots in residue rings modulo power of two. For probabilistic algorithms, the calculation complexity and the probability of successfully accomplishing the calculation in the first try are estimated.
The square roots extraction problem encountered in number-theoretic algorithms of cryptography used in combination with the Chinese remainder theorem for root square extraction in residue rings modulo an arbitrary number is considered. The article presents known probabilistic and deterministic algorithms for extracting square roots modulo a prime number, along with estimating the complexity of the algorithms and the probability of successfully completing the computation in the first try. New implementing in the elliptic curve cryptography algorithms for extracting square roots in finite fields of characteristic congruent to 3 modulo 4 are substantiated, and estimates of their complexity are given. Algorithms for extracting square roots in fields of even order are studied, and estimates of their complexity are given. Algorithms for extracting square roots in residue rings modulo prime powers and modulo power of two are demonstrated.

Key words: finite field, residue ring, square root extraction, square root modulo prime, square root modulo prime power, Hensel lifting, algorithm complexity estimation.

For citation: Gashkov S.B., Frolov A.B., Popova E.P. About the Complexity of Square Root Extraction Algorithms in Finite Fields and Residue Rings. MPEI Vestnik. 2018;5:79—88. (in Russian). DOI: 10.24160/1993-6982-2018-5-79-88.

Введение

Необходимость извлечения квадратных корней в числовых кольцах возникает при решении задач целочисленной факторизации, сложностью которых определяется безопасность криптографии с открытым ключом [1 — 5]. Применение эффективных алгоритмов извлечения квадратных корней в простых полях и кольцах по модулю степени простого числа позволяет по китайской теореме об остатках извлекать квадратные корни по модулю любого целого числа n, если известно его разложение

$$n = p_1^{e_1} \cdot p_2^{e_2} \cdot \cdots \cdot p_k^{e_k} \cdot \; i \neq j \rightarrow (p_i, p_j) = 1. \qquad (1)$$

Действительно, если $0 < a < n$ — квадратичный вычет по модулю n, то $a \bmod p^{e_i}$, $i = 1, ..., k$ является квадратичным вычетом по модулю p^{e_i}, то есть при существовании x, $x^2 \equiv a(\bmod n)$ существуют такие y_i, что $(\pm y_i)^2 \equiv a(\bmod p^{e_i})$, $i = 1, ..., k$.

При известных числах p^{e_i} числа y_i могут быть быстро найдены по алгоритмам, представленным в данной работе. Тогда из сравнений $x \equiv \pm y_i (\bmod p^{e_i})$, $i = 1, ..., k$ по китайской теореме об остатках можно вычислить квадратный корень из a по модулю n. Применительно к данному случаю согласно этой теореме при разложении (1) система сравнений $x \equiv a_i (\bmod p_i^e)$, $i = 1, ..., k$ имеет в интервале $[0, n-1]$ единственное решение x вида

$$x = \sum_{i=1}^{t} a_i N_i M_i \bmod n,$$

где $M_i = \dfrac{n}{p_i^{e_i}}$; $N_i = (M_i)^{-1} (\bmod\ p_i^{e_i})$, $i = 1, \ldots, k$.

Очевидно, что $(M_i, p_i^{e_i}) = 1$ и M_i можно найти, решая диофантово уравнение $N_i M_i + T p_i^{e_i} = 1$ посредством расширенного алгоритма Евклида.

Известно, что задача извлечения квадратного корня в кольце вычетов по составному модулю $n = pq$ (p, q — различные простые) эквивалентна по сложности проблеме факторизации этого модуля. Знание всех различных квадратных корней из элемента этого кольца позволяет разложить $n = pq$ на множители за полиномиальное время. Действительно,

$$s^2 - t^2 = (s + t)(s - t) \equiv 0(\bmod n),$$

что означает, что n делит $(s + t)(s - t)$. Но по выбору s и t число n не делит ни $(s + t)$, ни $(s - t)$. Отсюда, $s + t$ кратно числам p или q, и наибольший общий делитель (НОД) $(s + t, n)$ есть p или q. Поэтому, применяя алгоритм Евклида, можно разложить n.

В субэкспоненциальных алгоритмах факторизации для нахождения различных квадратных корней используются факторные базы, построение которых предполагает извлечение квадратных корней по модулю степени простого числа.

Изучены вероятностные и детерминированные алгоритмы извлечения квадратного корня из квадратичного вычета по модулю простого числа, т. е. в полях простой характеристики. Рассмотрены новые алгоритмы извлечения квадратных корней в полях характеристики p, $p \equiv 3(\bmod 4)$ и полях четного порядка, применяющиеся в эллиптической криптографии. Проанализированы особенности извлечения квадратных корней в полях четного порядка. Даны алгоритмы извлечения квадратного корня в числовых кольцах по модулю степени простого числа на основе принципа подъема решений Гензеля и обоснован новый алгоритм извлечения квадратных корней из вычетов, сравнимых с единицей по модулю 8, в кольцах вычетов по модулю степени двойки. Приведен новый рекурсивный алгоритм решения квадратных уравнений $x^2 - N = 0$, $N \equiv 1(\bmod 8)$ в кольце Z_{2^k} и показано существование четырех корней при $k \geq 3$. По результатам работы сделана итоговая таблица.

Квадратные корни в простых полях

Алгоритм Тонелли [2, 6]

Пусть p — простое число и $p - 1 = 2^s t$, где t — нечетное число; $s \geq 1$. Пусть α — порождающий элемент циклической группы Z_p^*, тогда $g = \alpha^t$ — порождающий элемент ее циклической подгруппы G порядка 2^s. Квадратичные вычеты g^{2i} по модулю p из G имеют порядки, равные степени 2:

$$\mathrm{ord} g^{2i} = \frac{\mathrm{ord} g}{(2^s, 2i)} = \frac{2^s}{2^m} = 2^{s-m},$$

где $2m = (2^s, 2i)$.

Обозначим Q_p и $\bar{Q}_p$ множества квадратичных вычетов и невычетов по модулю p. Пусть $a \in Q_p$, $a = a^{2c}$, тогда $a^t = \alpha^{2ct} = g^{2c} \in Q_p$ — квадратичный вычет из группы G. Элемент a^{-t}, обратный к a^t, также принадлежит G и, следовательно, существует четное k такое, что $a^{-t} = \alpha^{-2ct} = g^k$ ($k = -2c$ — четное, так как $-2c \equiv p - 1 - 2c(\bmod(p-1))$). Следовательно, существует четное k, $1 \leq k \leq 2^s$ такое, что

$$a^t g^k \equiv 1(\bmod p), \qquad (2)$$

где g — образующий элемент подгруппы G.

Определим

$$x = a^{(t+1)/2} g^{k/2}. \qquad (3)$$

Легко проверить, что $x^2 \equiv a(\bmod p)$.

Это справедливо при выборе любого образующего элемента группы G, поскольку любой такой элемент является t-й степенью подходящего образующего элемента α группы Z_p^*. Порождающими элементами группы G считаются все ее квадратичные невычеты $\alpha^{t(2r+1)}$, любой из которых можно использовать в (3), т. е. можно в качестве g взять t-ю степень любого квадратичного невычета $f = \alpha^{(2r+1)}$ из Z_p^*.

Таким образом, задача сведена к нахождению образующего элемента $g \in G$ и наименьшего четного k, удовлетворяющего соотношению (2).

Приведем алгоритм извлечения квадратного корня по модулю простого числа.

Образующий элемент группы G определяется случайным выбором элемента $f \in GF(p)^*$, проверкой $(f/p) = -1$ и принятием $g = f^t$ при ее положительном исходе. Вероятность успеха выбора такого элемента f — 1/2.

Для быстрого поиска k, удовлетворяющего (2), используем тот факт, что порядки квадратичных вычетов из G есть степени двойки.

Возьмем элемент

$$b = a^t \equiv a^t g^{2^s} (\mathrm{mod}\, p),$$

(здесь $g^{2^s} \equiv 1(\mathrm{mod}\, p)$) группы G.

Найдем наименьшее m, $0 \le m < s$ такое, что

$$b^{2^m} \equiv 1(\mathrm{mod}\, p). \quad (4)$$

Преобразуем b следующим образом:

$$b := bg^{2^{s-m}} \equiv a^t g^{2^{s-m}} (\mathrm{mod}\, p). \quad (5)$$

После этого порядок b уменьшается, но остается степенью 2:

$$b = \alpha^{2ct} g^{2^{s-m}} = \alpha^{2ct} \alpha^{t(2^{s-m})} = \alpha^{t(2(c+2^{s-m-1}))} = b^{2(c+2^{s-m-1})},$$

b остается квадратичным вычетом из G.

При повторном редуцировании m строго уменьшается. Когда m в (4) становится равным 0, b будет равно 1 и (5) преобразуется в (2).

Алгоритм Тонелли:

ВХОД: простое число p и целое $a \in Q_p$.

ВЫХОД: квадратный корень a по модулю p.

1. Представить $p - 1 = 2^s t$, где t нечетно; присвоить $b = a^t (\mathrm{mod}\, p)$; $r := s$, $k := 0$.

2. Найти квадратичный невычет $f \in \bar{Q}_p$; присвоить $g := f^t (\mathrm{mod}\, p)$

3. (Поиск экспоненты k);\\

пока $b \ne 1$:

3.1. Найти наименьшее неотрицательное m такое, что $b^{2^m} \equiv 1(\mathrm{mod}\, p)$.

3.2. Присвоить $b := bg^{2^{r-m}} \bmod p$; $k := k + 2^{r-m}$; $r := m$.

4. Вернуть $a^{(t+1)/2} g^{k/2} \bmod p$.

Поскольку $s < \log_2 p$ битовая сложность алгоритма, есть $O((\log p)^2 M(\log_2 p))$, где $M(n)$ — сложность умножения n-битных чисел.

Фюрер М. доказал, что $M(n) = n\log_2 n\psi(n)$, где $\psi(n)$ растет медленнее любой итерации логарифма [7 — 10].

Для средних значений n (порядка тысяч) быстрее алгоритм Шенхаге–Штрассена, а при малых n — метод Карацубы [2, 11].

Указанный алгоритм переносится на произвольные конечные поля $GF(q)$, $q = p^n$ и имеет сложность $O(\log_2 q)^2 M(GF(q))$, где $M(GF(q))$ — сложность умножения в поле $GF(q)$, если заранее известен невычет N в этом поле. Если он неизвестен, то его можно найти перебором, если быстро определять по элементу поля, является ли оно вычетом или невычетом по данному модулю. Для этого можно применить критерий Эйлера и алгоритм Брауэра для возведения в степень.

Доказано в предположении справедливости расширенной гипотезы Римана, что наименьший квадратичный невычет по модулю p не превышает $2\ln^2 p$ при $p > 1000$ [12]. Поэтому в поле $GF(p)$ его тоже можно найти со сложностью $O(\log_2 p)M(\log_2 p)$. Если не пользоваться недоказанными гипотезами, то указанный алгоритм следует рассматривать как вероятностный.

Очевидно, что $M(GF(q)) = O(M_{GF(p)}(n)M(\log_2 p))$, где $M_F(n)$ — верхняя оценка сложности умножения многочленов степени меньшей n над произвольным полем F, где под сложностью понимается число элементарных операций в поле F. Известно, что $M_F(n) = n\log_2 n\psi(n)$, где $\psi(n)$ растет медленнее любой итерации логарифма [13]. Для средних значений n (порядка тысяч) быстрее полиномиальный алгоритм Шенхаге, для которого $M_F(n) = 6n\log_2 n(\log_2 \log_2 n + O(1))$, при этом число умножений не больше $6n(\log_2 n + O(1))$ [14]. При малых n эффективнее полиномиальный алгоритм Карацубы [15]. Однако, если извлечение корня в данном поле нужно выполнять многократно (подобная ситуация встречается в теории кодирования и криптографии), то нужный для работы алгоритма квадратичный невычет можно вычислить один раз (предварительно) и сложностью его вычисления пренебречь. Тогда указанный алгоритм стоит рассматривать как детерминированный.

Некоторые детерминированные алгоритмы

Для случаев, когда $q \not\equiv 1(\mathrm{mod}\ 8)$ известны эффективные детерминированные алгоритмы без сделанных оговорок [1,6].

Если $p \equiv 3(\mathrm{mod}\, 8)$ или $p \equiv 7(\mathrm{mod}\, 8)$, то квадратный корень в поле $GF(q)$ можно получить по формуле $x = \pm a^{(q+1)/4}$. В этом случае $q + 1$ кратно 4. Пусть $x = a^{(q+1)/4}$, тогда с учетом того, что $a^{(q-1)/2} = 1$, имеем

$$x^2 = a^{(q+1)/2} = a^{(q-1)/2} a = a.$$

Если $q \equiv 5(\mathrm{mod}\, 8)$, то

а) $x = \pm a^{(q+3)/8}$, если $d = 1$;

б) $x = \pm(4a)^{(q+3)/8}/2$, если $d = q - 1$, где $d = a^{(q-1)/4}$.

В этом случае $q + 3$ кратно 8, поскольку $(q-1)/2$ четно, то -1 по критерию Эйлера — квадратичный вычет.

Пусть $x = a^{(q+3)/8}$, $a \in Q_q$. Из $a^{(q-1)/2} = 1$, с учетом того, что 1 в $GF(q)^*$ имеет только два квадратных корня 1 и -1, получим $a^{(q-1)/4} = \pm 1$, следовательно,

$$x^2 = a^{(q+3)/4} = a^{(q-1)/4} a = \pm a.$$

Если $x^2 = a$, то решением является $x = a^{(q+3)/8}$, т. е. вариант a. Если $x^2 = -a$, то $-x^2 = (\sqrt{-1}x)^2 = a$. Поэтому решение — $x = \sqrt{-1}a^{(q+3)/8}$ и задача сведена к вычислению $\sqrt{-1}$. В качестве $\sqrt{-1}$ можно взять $b^{(q-1)/4}$, где b — квадратичный невычет в поле $G_F(q)$, так как по критерию Эйлера $(b^{(q-1)/4})^2 = b^{(q-1)/2} = -1$.

В случае поля $GF(p)$ в этом качестве можно взять квадратичный невычет $b = 2$. Тогда двойка — квадратичный невычет по модулю p по свойству символа Лежандра:

$$p^2 - 1 = (p+1)(p-1) = (8k+6)(8k+4) = 8(4k+3)(2k+1),$$

а правая часть является нечетным числом, умноженным на 8, поэтому $2 \in \overline{Q}_p$ и можно использовать $2^{(p-1)/4}$ вместо –1. Таким образом,

$$x = \sqrt{-1}a^{(p+3)/8} = 2^{(p-1)/4}a^{(p+3)/8} = (4a)^{(p+3)/8}/2,$$

т.е. решением является $x = \pm(4a)^{(q+3)/8}/2$, т. е. вариант *б*.

Алгоритм извлечения квадратного корня по модулю p, $p \equiv 3 (\mathrm{mod}\, 8)$ или $p \equiv 7 (\mathrm{mod}\, 8)$, или $p \equiv 5 (\mathrm{mod}\, 8)$:

ВХОД: степень простого числа p, $p \equiv 3 (\mathrm{mod}\, 8)$ или $p \equiv 7 (\mathrm{mod}\, 8)$, *или* $p \equiv 5 (\mathrm{mod}\, 8)$, квадратичный вычет a в поле $G_F(p)$.

ВЫХОД: квадратный корень x из элемента a.

1. Если $p \equiv 3 (\mathrm{mod}\, 8)$ или $p \equiv 7 (\mathrm{mod}\, 8)$ вернуть $a^{(p+1)/4}$.
2. Если $a^{(p-1)/4} = 1$, вернуть $a^{(p+3)/8}$.
3. Вернуть $(4a)^{(p+3)/8}/2$.

Сложность этого алгоритма $O((\log p)M(\log p))$.

Рассмотренные алгоритмы могут быть распространены на любые конечные поля $GF(q)$ нечетного порядка $q = p^m$, p — простое, $m \geq 1$.

Алгоритм Чиполлы

Данный алгоритм основан на идее, опубликованной итальянским математиком Мишелем Чипполла (Michele Cipolla) в 1907 г., и работает несколько быстрее алгоритма Тонелли [1, 16].

Пусть надо извлечь квадратный корень по модулю p из данного числа a. Тогда его можно рассматривать как квадратичный вычет в поле $GF(p)$, т. е. символ Лежандра $(a/p) = 1$. Выберем случайное целое число t в отрезке от 0 до $p - 1$, такое, что $t^2 - a$ будет квадратичным невычетом, т. е. символ Лежандра $\left(\dfrac{t^2-a}{p}\right) = -1$.

Лемма 1. *Вероятность удачного выбора числа t равна* $(p - 1)/(2p)$.

Повторим выбор числа t до тех пор, пока не найдем требуемое. С высокой вероятностью успех будет достигнут не более чем на десятой попытке. Применим для вычисления символа Лежандра алгоритм Якоби–Кронекера, тогда оценка сложности процедуры поиска подходящего t с высокой вероятностью будет равна $O(\log_2 p)^2$.

Построим квадратичное расширение поля $GF(p)$ путем присоединения к нему корня α неприводимого квадратного двучлена $x^2 - (t^2 - a)$, тогда элементы указанного расширения выглядят так $a + b\alpha$, где $a, b \in GF(p)$. Операция умножения определяется формулой $(a + b\alpha)(c + d\alpha) = ac + bd\alpha^2 + (ad + bc)\alpha$, где $\alpha^2 = t^2 - a \in GF(p)$. Сложение выглядит как $(a + b\alpha) + (c + d\alpha) = (a + c) + (b + d)\alpha$.

Лемма 2. *Указанное множество относительно определенных выше операций образует поле, изоморфное полю* $GF(p^2)$, *умножение в котором можно свести к четырем умножениям в поле* $GF(p)$. *Если обозначить* $M(n)$ *сложность умножения двух n-разрядных чисел, то сложность умножения в поле* $GF(p^2)$ *асимптотически будет не больше* $12M(\log_2 p)$, *а для простых чисел специального вида* $p = 2^k \pm c$ *при малых c асимптотически не больше* $4M(\log_2 p)$.

Возьмем $x = (t + \alpha)^{(p+1)/2} \in GF(p^2)$, тогда $x \in GF(p)$ и $x^2 = a$.

Лемма 3. *Справедливы следующие равенства в поле* $GF(p^2)$:

$$\alpha^{p-1} = (t^2 - a)^{(p-1)/2} = -1,\ \alpha^p = -\alpha, \alpha^{p+1} = -\alpha^2;$$
$$t^{p+1} = t^2, t^p = t, x^2 = (t+\alpha)^{p+1} =$$
$$= t^{p+1} + (p+1)t^p\alpha + (p+1)t\alpha^p + \alpha^{p+1} =$$
$$= t^2 + t\alpha - t\alpha - \alpha^2 = t^2 - (t^2 - a) = a.$$

Действительно, в формуле бинома $(a + b)^{p+1}$ все коэффициенты, кроме крайних, делятся на p. Остается заметить, что $x \in GF(p)$, так как все корни уравнения $x^2 = a$ лежат в поле $GF(p)$.

Оценка сложности алгоритма Чипполы вытекает из Леммы 4.

Лемма 4. *Сложность вычисления элемента* $x = (t + \alpha)^{(p+1)/2}$ *оценивается асимптотически как* $12l((p + 1)/2)M(\log_2 p) \sim 12M(\log_2 p)\log_2 p$, *а для чисел p специального вида как* $4M(\log_2 p)\log_2 p$, *где* $l(n)$ *— длина кратчайшей аддитивной цепочки для n.*

Выяснить, существует ли квадратный корень из заданного элемента поля нечетного порядка $GF(q)$ можно еще быстрее. В [16] это сделано с помощью детерминированного алгоритма сложности $O(\log_2 q)^2$. Известно, что в случае простого p это выполняется вычислением символа Лежандра с помощью быстрой версии алгоритма Евклида со сложностью $O(\log_2 \log_2 pM(\log_2 p))$.

Быстрое извлечение корней в небинарных полях

В поле $GF(p^r)$ при $p \equiv 3 (\mathrm{mod}\, 4)$ и нечетном r квадратный корень можно быстро вычислить благодаря лемме 5.

Лемма 5. *При* $q \equiv 3 (\mathrm{mod}\, 4)$ *квадратный корень в поле* $GF(q)$ *вычисляется по формуле* $\sqrt{x} = x^{(q+1)/4}$. *В частности, это верно при* $q = p^r$, *где* $p \equiv 3 (\mathrm{mod}\, 4)$, *а r — нечетно. Если выбрать в* $GF(p^r)$ *полиномиальный базис, то число операций в поле* $GF(p)$ *для извлечения квадратного корня в поле* $GF(p^r)$ *равно* $O((\log_2 r)M(r)) + r^{1+o(1)}$. *Если базис определяется неприводимым биномом или является оптимальным нормальным базисом, то слагаемое* $r^{1+o(1)}$ *можно отбросить.*

В поле $GF(p^s)$, где $p \equiv 3 (\mathrm{mod}\, 4)$, s — четно, этот алгоритм не работает, а алгоритм Чиполлы имеет сложность $O((\log_2 s)M(s)) + s^{1+o(1)}$ операций в поле $GF(p)$.

Пусть $s = 2k^r$, r — нечетно. Покажем, что при k, достаточно большом в сравнении с r, сложность извлечения квадратного корня по порядку равна сложности умножения в поле $GF(p^s)$. Рассмотрим только случай $p = 3$.

Следуя [14, 17], в поле $GF(p^s)$ построим башню (последний этаж башни — в поле $GF(p^{2s})$), состоящую из квадратичных расширений подполей

$$GF(3^r) \subset GF(3^{2r}) \subset \ldots \subset GF(3^{2^k r}) \subset GF(3^{2^{k+1} r}),$$

на первом этаже которой выбирается базис $\{1, \imath\}$, где $\imath \in GF(3^{2r}) \setminus GF(3^r)$, $\imath^2 = -1$, а на каждом следующем $GF\left(3^{2^{i-1} r}\right) \subset GF\left(3^{2^i r}\right)$ — базис $\{1, \omega_i\}$ где

$$\omega_i \in GF(3^{2^i r}) \setminus GF(3^{2^{i-1} r}), \quad \omega_1 = 1 - \imath.$$

Можно доказать по индукции возможность такого выбора базисов в этой башне. Действительно, $1 - \imath$ не квадрат в $GF(3^{2r})$, $\omega_i^{2^{i-1}} = \omega_1$, $\omega_i^{2^{i+1}} = -1$, $(3^{2^i r} - 1) = (3^{2r} - 1)(3^{2r} + 1)(3^{4r} + 1)\ldots(3^{2^{(i-1)} r} + 1)$ делится на 2^{i+2} и не делится на 2^{i+3}, поэтому $\omega_i^{(3^{2^i r} - 1)/2} = (-1)^{(3^{2^i r} - 1)/2^{i+2}} = -1$, значит $\omega_i \in GF(3^{2^i r})$, согласно критерию Эйлера, не является квадратом в этом поле, следовательно $\omega_{i+1} = \sqrt{\omega_i} \in GF(3^{2^{i+1} r}) \setminus GF(3^{2^i r})$. В качестве базиса в поле $GF(3^{2^i r})$ над подполем $GF(3^r)$ можно взять произведение базисов

$$\{1, \omega_i\} \times \{1, \omega_{i-1}\} \times \ldots \times \{1, \omega_2\} \times \{1, \imath\},$$

а над подполем $GF(3^{2r})$ — $\{1, \omega_i\} \times \{1, \omega_{i-1}\} \times \ldots \times \{1, \omega_2\}$.

Из равенства $\omega_i^{2^{i-1}} = \omega_1$ и свойств двоичной системы счисления следует, что последний базис является перестановкой полиномиального базиса $\{1, \omega_i, \ldots, \omega_i^{2^{i-1} - 1}\}$, где $\omega_i^{2^{i-1}} = \omega_1 = 1 - \imath \in GF(3^{2r})$. Поэтому умножение в указанном базисе поля $GF(3^{2^i r})$ над подполем $GF(3^{2r})$ сводится к умножению многочленов степени $2^{i-1} - 1$ и приведению полученного многочлена степени $2^i - 2$ по модулю многочлена $x^{2^{i-1}} - \omega_1$.

Под троичными операциями (можно выразить через двоичные, причем разными способами) понимаются операции в поле $GF(3)$, а под троичными многочленами — многочлены с коэффициентами в поле $GF(3)$. Сложность умножения троичных многочленов степени, меньшей n, обозначим $M(n)$ а сложность умножения многочленов над полем $GF(3^r)$ — $M_r(n)$, но далее индекс r опустим. Предположим, что $M(2n) \geq 2M(n)$.

Лемма 6. *Число троичных операций для умножения в построенном базисе поля $GF(3^s)$, $s = 2^k r$ равно $O(M(2^k)M(r))$. Число троичных операций для возведения в степени $(3^s - 1)/2$ и $(3^s + 1)/4$ равно $O(M(2^k)M(r)) + r^{1+o(1)}$. Если базис определяется неприводимым биномом или является оптимальным нормальным базисом, то слагаемое $r^{1+o(1)}$ можно отбросить.*

Для доказательства заметим, что для произвольного элемента $a \in GF(3^s)$ возведение в куб, согласно

$$(a_0 + a_1\omega_k + \ldots + a_{2^{k-1}-1}\omega_k^{2^{k-1}-1})^3 = a_0^3 + a_1^3\omega_k^3 + \ldots + a_{2^{k-1}-1}^3\omega_k^{(2^{k-1}-1)3},$$

сводится к 2^{k-1} возведениям в куб в подполе $GF(3^{2r})$ и не более чем 2^k умножениям в этом подполе на элементы $\omega_1 = 1 - i$ и $\omega_1^2 = i$.

Аналогично операция Фробениуса возведения в произвольную степень 3^m сводится к 2^{k-1} возведениям в степень 3^m в подполе $GF(3^{2r})$ и не более чем 2^k умножениям в этом подполе на элементы $1 - i$, i, -1, $i - 1$, $-i$. Для возведения x в степень $(3^s + 1)/4$ (при нечетном s) возводим в степень $(3^{s-1} - 1)/4$, потом в куб, и умножим на x. Для возведения в степени $(3^s - 1)/2 = (3^s - 1)/(3 - 1)$ и $(3^s - 1)/4$ (при четном s) сначала возводим в степень $(3^s - 1)/(3^{2r} - 1)$, а потом в степень $(3^{2r} - 1)/2$ (или $(3^{2r} - 1)/4$) с помощью метода Брауэра [18]. Для этого достаточно не более $O(\log_2 r)$ умножений и операций Фробениуса возведения в степени 3^{m_i}, где $\sum_i m_i = O(r)$.

Обозначим $F(s)$ сложность вычисления преобразования Фробениуса (возведения в степень вида 3^n) в поле $GF(3^s)$, $s = 2^k r$. Если в подполе $GF(3^r)$ существует оптимальный нормальный базис [19], то $F(s) = O(s)$. Оценка $F(s) = O(s)$ справедлива также, если в поле $GF(3^r)$ выбрать полиномиальный базис над подполем $GF(3^{r/t})$, соответствующий неприводимому биному $x^t - a$, $a \in GF(3^{r/t})$. В обоих указанных случаях сложность возведения в степень $(3^{2r} - 1)/2$ оценивается как $O((\log_2 r)M(2^k)M(r))$. При произвольном r в поле $GF(3^r)$ можно выбрать полиномиальный базис, определяемый неприводимым многочленом, содержащим не более пяти одночленов. Тогда, для возведения в степень 3^m справедливо равенство $F(s) = O(ms) = O(rs)$, а сложность возведения в степень $(3^{2r} - 1)/2$ равна $O((\log_2 r)M(2^k)M(r))$. Без указанного предположения для выполнения произвольной операции Фробениуса в подполе $GF(3^r)$ можно воспользоваться модулярной суперпозицией многочленов. Используя [20], покажем, что $F(r) = r^{1+o(1)}$. Тогда $F(s) = 2^k r^{1+o(1)} = sr^{o(1)}$ и сложность возведения в степень $(3^{2r} - 1)/2$ оценивается как $O((\log_2 r)M(2^k)M(r)) + sr^{o(1)}$.

Для возведения в степень $(3^s - 1)/2$, $s = 2^k r$ сначала найдем $x_1 = x^{q^{2^{k-1}} + 1} = x^{q^{2^{k-1}}} x$, где $q = 3^r$.

В силу малой теоремы Ферма $x_1 \in GF(q^{2^{k-1}}) \subset GF(3^s)$. Аналогично $x_i = x_{i-1}^{q^{2^{k-i}} + 1} \in GF(q^{2^{k-i}})$, $i = 1, \ldots, k - 1$, при этом $x_{k-1} \in GF(q^2)$, а так как

$$(q^{2^{k-1}} + 1)(q^{2^{k-2}} + 1)\ldots(q^2 + 1) = (q^{2^k} - 1)/(q^2 - 1),$$

то $x_{k-1} = x^{(q^{2^k} - 1)/(q^2 - 1)}$.

Возведя $x_{k-1} \in GF(q^2)$ в степень $(q^2 - 1)/2$, найдем $x^{(q^{2^k} - 1)/2} = x^{(3^s - 1)/2}$ (аналогично вычисляется $x^{(3^s - 1)/4}$).

Поскольку $x_i \in GF\left(q^{2^{k-i}}\right)$ (где $x_0 = x$), то

$$x_i = u_i + v_i\omega_{k-i}, u_i, v_i \in GF\left(q^{2^{k-i-1}}\right),$$

поэтому

$$x_i^{q^{2^{k-i-1}}} = u_i^{q^{2^{k-i-1}}} + v_i^{q^{2^{k-i-1}}}\omega_{k-i}^{q^{2^{k-i-1}}} = u_i + v_i\omega_{k-i}^{q^{2^{k-i-1}}}.$$

Значит, для вычисления

$$x_{i+1} = x_i^{q^{2^{k-i-1}}} x_i = (u_i + v_i\beta_{k-i}\omega_{k-i})(u_i + v_i\omega_{k-i}) =$$
$$= (u_i^2 + v_i^2\beta_{k-i}\omega_{k-i-1}) + (u_iv_i(1+\beta_{k-i}))\omega_{k-i}$$

достаточно выполнить пять умножений и два сложения в поле $GF(q^{2^{k-i-1}})$. Используя оценку $O(M(2^k)M(r))$ для умножения в поле $GF(q^{2^k})$, получим для возведения в степень $(q^{2^k}-1)/(q^2-1)$ в этом поле такую же по порядку оценку.

Остается воспользоваться тем, что сложность возведения в степень $(q^2-1)/2 = (3^{2r}-1)/2$ в поле $GF(q^2)$ оценивается как $O(\log_2 r(M(2^k)M(r))) + r^{1+o(1)}$, а в предположении, что в поле $GF(q)$ выбран оптимальный нормальный или «хороший» полиномиальный базис, последнее слагаемое можно убрать.

Подобно [17] доказываются лемма 7 и теорема.

Лемма 7. *Сложность вычисления мультипликативного обратного в том же базисе в поле $GF(3^s)$ равна $O(M(2^k)M(r)) + r^{1+o(1)}$. Если выбранный в базис $GF(3^r)$ определяется неприводимым биномом или является оптимальным нормальным базисом, то слагаемое $r^{1+o(1)}$ можно заменить на $O(\log_2 rM(r))$.*

Теорема. *Извлечение квадратного корня в поле $GF(3^s)$, $s = 2^k r$ можно выполнить со сложностью $O(M(2^k)M(r) + \log_2 rM(r)) + kr^{1+o(1)}$. Если базис в поле $GF(3^r)$ определяется неприводимым биномом или является оптимальным нормальным базисом, то слагаемое $kr^{1+o(1)}$ можно отбросить.*

Для доказательства заметим, что извлечение корня в поле $GF(3^{2^k r})$ при использовании базиса $\{1, \omega_i\}$ сводится к извлечению корня в подполе $GF(3^{2^{k-1}r})$. Пусть $q = 3^{2^{i-1}r}$, $GF(3^{2^{i-1}r}) = GF(q)$, $GF(3^{2^{i-1}r}) = GF(q^2)$ и $x \in GF(q^2) \setminus GF(q)$. Для нахождения $y \in GF(q^2)$ такого, что $y^2 = x$, рассмотрим $z = y + y^q$. Так как $z^q = y^q + y^{q^2} = y^q + y = z$, то $z \in GF(q)$, значит $w = z^2 \in GF(q)$. Но

$$w = z^2 = (y(1+y^{q-1}))^2 = y^2(1+x^{(q-1)/2})^2 = x(1+x^{(q-1)/2})^2,$$

значит

$$z = \pm\sqrt{w} = \pm\sqrt{x}(1+x^{(q-1)/2}) = \pm y(1+x^{(q-1)/2}),$$

откуда

$$y = \frac{\pm z}{1+x^{(q-1)/2}} = \frac{\pm\sqrt{w}}{1+x^{(q-1)/2}} = \frac{\pm\sqrt{x(1+x^{(q-1)/2})^2}}{1+x^{(q-1)/2}}.$$

Для вычисления $y = \sqrt{x}$ по указанной формуле нужно выполнить возведение x в степень $(q-1)/2$, сложение с 1, возведение результата в квадрат, умножение на x (в результате чего получается $w \in GF(q)$, а значит оно проще произвольного умножения в поле $GF(q^2)$), извлечение корня в поле $GF(q)$ и деление на ранее вычисленный элемент $1 + x^{(q-1)/2}$. Если он равен нулю, выбираем $c \in GF(q^2) \setminus GF(q)$ и вычисляем указанным методом корень из c^2x, а потом полученный результат делим на c. При этом вместо $1 + x^{(q-1)/2} = 0$ делить надо будет на $1 + c^{q-1}x^{(q-1)/2} \neq 0$, так как $c^{q-1} \neq 1$.

В качестве c возьмем ω_i, тогда умножение выполняется на ω_{i-1}, то есть со сложностью $O(r)$, а деление на ω_i со сложностью $O(M(2^i)M(r))$. Сложность всех операций, кроме возведения в степень равна $O(M(2^i)M(r))$, а сложность возведения в степень согласно лемме 2 равна $O(M(2^i)M(r)) + r^{1+o(1)}$. Если обозначить $S(i)$ сложность извлечения корня в поле $GF(3^{2^i r})$, то

$$S(i) \le S(i-1) + O(M_r(2^i)M(r)) + r^{1+o(1)};$$
$$S(1) = O((\log_2 r)M(r)) + r^{1+o(1)},$$

значит $S(k) = O(M(2^k)(M(r))) + kr^{1+o(1)}$.

Если корень из $x \in GF(3^s)$ в $GF(3^s)$ не существует, то он принадлежит $GF(3^{2s})$. Для его вычисления применяется тот же алгоритм, только начинать его надо на этаже $GF(3^{2^{k+1}r})$.

Приведем пример вычисления квадратного корня данным алгоритмом.

Пусть нужно извлечь корень из элемента

$$a = -\omega_3^3 - \imath\omega_3 - 1 = -\omega_3(\omega_2 + \imath) - 1 \in GF(3^8).$$

Вычисляем

$$b = (a^9 \cdot a)^3 \cdot a^9 \cdot a + 1 = a^{40} + 1 =$$
$$= -\omega_3^3 + (1+\imath)\omega_3^2 + \imath\omega_3 - \imath \in GF(3^8);$$
$$c = b^2 \cdot a = -(1+\imath)\omega_2 - \imath \in GF(81),$$

тогда $\sqrt{a} = \sqrt{c}/b$.

Для расчета $\sqrt{c}$ найдем

$$b_1 = c^4 + 1 = c^3 \cdot c + 1 = -\imath\omega_2 - 1 - \imath \in GF(81);$$
$$c_1 = b_1^2 \cdot c = 1 \in GF(9),$$

и $\sqrt{c_1} = 1$, значит

$$\sqrt{c} = \sqrt{c_1}/b_1 = 1/b_1 =$$
$$= -(1+\imath)\omega_2 - 1, 1/b = -\imath\omega_3^3 - 1 = -\imath\omega_2\omega_3 - 1;$$
$$\sqrt{a} = \sqrt{c}/b = ((1+\imath)\omega_2 + 1)(\imath\omega_2\omega_3 + 1) =$$
$$= \imath\omega_3^3 + (1+\imath)\omega_3^2 - \imath\omega_3 + 1.$$

Быстрое извлечение корней в полях четного порядка

Квадратные корни в полях четного порядка легко вычисляются, благодаря следующему факту: каждый

элемент $a \in GF(2^n)$ имеет единственный квадратный корень $a^{2^{n-1}}$. Если в поле выбран любой нормальный базис, то операция Фробениуса возведения в степень 2^k имеет нулевую битовую сложность, так как сводится к циклическому сдвигу. В произвольном полиномиальном базисе $\{1, \alpha, ..., \alpha^{n-1}\}$ поля $GF(2^n)$ вычисление преобразования Фробениуса, а значит и извлечение квадратного корня согласно [20], выполняется со сложностью $n^{1+o(1)}$.

Если $\sqrt{\alpha} \in GF(2^n)$ вычислен заранее, то при $n = 2m + 1$ имеем

$$\sqrt{a_0 + a_1\alpha + \ldots + a_{n-1}\alpha^{n-1}} = \\ = a_0 + a_2\alpha + \ldots + a_{2m}\alpha^m + \sqrt{\alpha}(a_1 + a_3\alpha + \ldots \\ \ldots + a_{2m+1}\alpha^m)$$

и вычисление квадратного корня выполняется со сложностью $O(M_{GF(2)}(n))$ (случай $n = 2m$ аналогичен). Если $\sqrt{\alpha}$ в рассматриваемом базисе имеет ограниченное число ненулевых (единичных) коэффициентов, то приведенную оценку можно заменить на $O(n)$. В частности, если в рассматриваемом базисе $f(\alpha) = 0$, где неприводимый двоичный многочлен $f(x) = x^n + x^k + 1$, то согласно [21] при четном n, а также нечетных n и k число единичных коэффициентов в $\sqrt{\alpha}$ не больше 3. Если n нечетно и k четно, то можно взять взаимный многочлен $x^n + x^{n-k} + 1$, который будет неприводимым и $n - k$ у него будет нечетным. Значит всегда при существовании неприводимого двоичного трехчлена степени n, его можно выбрать так, что в соответствующем базисе число единичных коэффициентов в $\sqrt{\alpha}$ будет не больше 3, поэтому сложность извлечения квадратных корней в поле $GF(2^n)$ в подходящем полиномиальном базисе равна $O(n)$. Экспериментально установлено, что для около 80% значений n — неприводимые трехчлены степени n, значит указанный алгоритм имеет сложность $O(n)$.

Извлечение квадратного корня по модулю, равному степени простого числа

Квадратные корни по модулю степени нечетного простого числа

В случае модуля, равного степени простого числа, известен быстрый алгоритм извлечения квадратного корня [1].

Для решения сравнений по модулю p^n используется метод «подъема решений» (или «лифт») немецкого алгебраиста К. Гензеля, использованный им при построении теории p-адических чисел.

Известна следующая модификация для извлечения квадратных корней. Пусть p — нечетное простое число и $t_1 = t_0 + t_1 p + \cdots t_{\beta-1} p^{\beta-1}$ — решение сравнения

$$t_1^2 \equiv (t_0 + t_1 p + \cdots t_{\beta-1} p^{\beta-1})^2 \equiv a (\mathrm{mod} p^{\beta}), \qquad (6)$$

тогда $t_0^2 \equiv a(\mathrm{mod} p)$.

Обозначим число $b_0 = t_0$ а $b_i = t_0 + t_1 p + \cdots + t_i p^i$. Заметим, что $b_{i-2} + t_{i-1} p^{i-1} = t_0 + t_1 p + \cdots + t_{i-1} p^{i-1}$— решение сравнения $(b_{i-2} + t_{i-1} p^{i-1})^2 \equiv a(\mathrm{mod} p^i)$.

Отсюда по формуле квадрата суммы с учетом, что $p^{2(i-1)} \equiv 0(\mathrm{mod} p^i)$, $2t_{i-1} b_{i-2} p^{i-1} \equiv (a - b_{i-2}^2)(\mathrm{mod} p^i)$, следовательно

$$2b_{i-2} t_{i-1} \equiv \frac{a - b_{i,i-2}^2}{p^{i-1}} (\mathrm{mod} p).$$

Таким образом,

$$t_{i-1} \equiv \frac{a - b_{i,i-2}^2}{p^{i-1}} 2^{-1} b_{i-2}^{-1} (\mathrm{mod} p).$$

Алгоритм вычисления квадратного корня из квадратичного вычета a по модулю степени p^{β} простого числа p:

0. Разложить число a по степеням простого числа p и пополнить полученный список *decomp* старшим коэффициентом 0 длины len(*decomp*) принять β = len(*decomp*) и составить список *rightparts* = $= [a \bmod p, a \bmod p^2, \ldots, a \bmod p^{\beta}]$ правых частей сравнений (5).

Вычислить список $stepsofp = [1, p, p^2, \ldots, p^{\beta+2}]$ степеней простого числа p.

1. Вычислить $t = \sqrt{rightparts[0]} \bmod p$.
   Принять $b = t$.
2. Для $i = 1, \beta$:

$$t = \frac{(rightparts[i] - b^2) \bmod (stepsofp)[i]}{stepsofp[i-1]} 2^{-1} b^{-1} \bmod p;$$

$tstepsofp = t \cdot stepsofp[i]$;

$b = b + tstepsofp \bmod stepsofp[i+1]$.

3. Вернуть b.

Использованный прием вычисления решения в порядке возрастания показателя степени одного из параметров и есть подъем решения Гензеля. Этот алгоритм применим и для решения произвольных алгебраических уравнений вида $f(x) = 0$ в кольце Z_{p^k}, $k \in N$, p — простое число.

Шаг 1. Найти все решения уравнения $f(x) = 0$ в кольце Z_p, если они существуют. Если их нет, то нет решений и в любом кольце Z_{p^k}.

Шаг 2. Для каждого решения a_0, найденного на первом шаге, и всех $i \in \{1, 2, ..., k-1\}$ найти решения линейного уравнения

$$f'(a_0) a_i = \frac{f(a_0 + a_1 p + \ldots + a_{i-1} p^{i-1})}{p^i} (\mathrm{mod} p), \qquad (7)$$

удовлетворяющие условию $a_i \in \{0, 1, ..., p-1\}$.

Если это уравнение не имеет решений для какого-то i, то перейти к другому a_0 и повторить шаг 2.

Если же уравнение (6) имеет решения для всех i, $i \in \{1, ..., k-1\}$, то числа вида

$$a = \sum_{i=0}^{k-1} a_i p^i$$

являются решениями исходного уравнения $f(x) = 0$ в кольце Z_{p^k}.

Этот алгоритм неприменим в случае $p = 2$, так как $2^{-1}\text{mod}2$ не существует.

Сложность алгоритма определяется сложностями первого шага и последующего подъема решения (шаг 2), $O(nM(log_2 p)M_p(n)$, где $M_p(n)$ — сложность умножения n-разрядных чисел в p-ичной системе счисления; $M(\log_2 p)$ — сложность умножения в двоичной системе чисел, меньших, чем p.

Извлечение квадратного корня в кольцах по модулю степени двойки

Рассмотрим уравнение

$$x^2 \equiv n(\text{mod}2^\beta) \quad (8)$$

при нечетном n, заметим, что его решения существуют только при $n \equiv 1(\text{mod}8)$.

Утверждение. Если $n \equiv 1(\text{mod}8)$, то уравнение (8) имеет 4 корня

$$x_1^{(\beta)}, x_2^{(\beta)} = 2^{\beta-1} - x_1^{(\beta)}, x_3^{(\beta)} = 2^{\beta-1} + x_1^{(\beta)}, x_4^{(\beta)} = 2^\beta - x_1^{(\beta)}, \quad (9)$$

при этом если $(x_1^{(\beta-1)})^2 \text{mod}2^{(\beta)} = n \,\text{mod}2^{(\beta)}$, то $x_1^{(\beta)} = x_1^{(\beta-1)}$, иначе $x_1^{(\beta)} = x_2^{(\beta-1)}$.

Здесь $x_1^{(\beta-1)}$, $x_2^{(\beta-1)}$ — соответствующие корни уравнения

$$x^2 \equiv n(\text{mod}\, 2^{\beta-1}). \quad (10)$$

Доказательство (индукция по β). При $\beta = 3$

$$x_1^{(3)} = 1, x_2^{(3)} = 2^2 - 1 = 3, x_3^{(3)} = 2^2 + 1 = 5, x_4^{(3)} = 2^3 - 1 = 7.$$

Легко проверить, что если

$$x_1^{(\beta-1)}, x_2^{(\beta-1)} = 2^{\beta-2} - x_1^{(\beta-1)}, x_3^{(\beta-1)} = 2^{\beta-2} + x_1^{(\beta-1)}, x_4^{(\beta-1)} = 2^{\beta-1} - x_1^{(\beta-1)}$$

— корни уравнения (10) и $x_1^{(\beta-1)2} \equiv n(\text{mod}2^\beta)$ ($x_1^{(\beta-1)}$ является и корнем уравнения (8)), то при $x_1^\beta = x_1^{(\beta-1)}$ элементы (9) — корни уравнения (8). Если же $(x_1^{(\beta-1)})^2 \not\equiv n(\text{mod}2^\beta)$ ($x_1^{(\beta-1)}$ не является корнем уравнения (8), в этом случае $2^{\beta-1} + (x_1^{(\beta-1)})^2 \equiv n(\text{mod}2^\beta)$), то корнем уравнения (8) будет корень $x_2^{\beta-1}$ уравнения (10):

$$(x_2^{(\beta-1)})^2 = (2^{\beta-2} - x_1^{(\beta-1)})^2 = 2^{2\beta-4} - 2^{\beta-1}x_1^{(\beta-1)} + (x_1^{\beta-1})^2 \equiv -2^{\beta-1} + (x_1^{\beta-1})^2 \equiv 2^{\beta-1} + (x_1^{\beta-1})^2 (\text{mod}2^\beta).$$

В данном случае учтено, что $x_1^{(\beta-1)}$ — четное, тогда В этом случае элементы (8) являются корнями уравнения (8) при $x_1^{(\beta)} = x_2^{(\beta-1)}$.

Таким образом, обоснован следующий алгоритм решения уравнения (7):

принять $x = 1$;

для $i = 4, \ldots, \beta$:

если $x^2 \text{mod}2^i \neq n \,\text{mod}(2^i)$, принять $x = 2^{i-1} - x$;

вернуть $x, 2^{i-1} - x, 2^{i-1} + x, 2^i - x$.

Сложность этого алгоритма оценивается как $O(n \cdot M(n))$.

Обоснованные оценки сложности извлечения квадратного корня в конечных полях приведены в таблице.

Работа выполнена при поддержке РФФИ (проект № 17-01-00485а).

Литература

1. **Коблиц Н.** Курс теории чисел и криптографии. М.: Науч. Изд-во НТП, 2001.

2. **Василенко О.Н.** Теоретико-числовые алгоритмы в криптографии. Изд-во МЦНМО, 2003.

3. **Глухов М.М., Круглов И.А., Пичкур А.Б., Черемушкин А.В.** Введение в теоретико-числовые методы криптографии. СПб. — М. — Краснодар: Изд-во Лань, 2011.

4. **Болотов А.А., Гашков С.Б., Фролов А.Б. Часовских А.А.** Элементарное введение в эллиптическую криптографию. Кн. 1. Алгебраические и алгоритмические основы. М.: Изд-во Ленанд, 2019.

5. **Кредалл Р., Померанс К.** Простые числа. Криптографические и вычислительные аспекты. М.: Изд-во «Книжный дом «Либерком», 2011.

6. **Венбо Мао.** Современная криптография. М.: Издат. дом Вильямс, 2005.

7. **Furer M.** Faster Integer Multiplication // SIAM J. Computing. 2009. Iss. 3. V. 39. Pp. 979—1005.

8. **Saha A., C., Kurur P., Saptharishi R.** Fast Integer Multiplication Using Modular Arithmetic // SIAM J. Computing. 2013. Iss. 2. V. 42. Pp. 685—699.

9. **Harvey D., van der Hoeven J., Lecerf G.** Even Faster Integer Multiplication // J. Complexity. 2016. V. 36. Pp. 1—30.

10. **Covanov S., Thome E.** Fast Integer Multiplication Using Generalized Fermat Primes // J. Mathematics of Computation. 2016. V. 1. Pp. 1—27.

11. **Bernstein D.J. Chuengsatiansup C., Lange T.** Curve41417:Karatsuba Revisited // Proc. Cryptographic Hardware and Embedded Systems. 2014. Pp. 316—334.

12. **Bach E.** Explicit Bounds for Primality Testing and Related Problems // Math.Comp.1989. No. 55. Pp. 355—380.

13. **Harvey D.,van der Hoeven J., Lecerf G.** Faster Polynomial Multiplication Over Finite Fields // ArXive. arXiv:1407.3361. 2014.

14. **Гашков С.Б., Сергеев И.С.** О сложности и глубине булевых схем для умножения и инвертирования в

Сложность извлечения квадратного корня в конечных полях

Алгоритм	Вероятность выбора нужного элемента с первой попытки	Сложность выбора нужного элемента	Сложность детерминированной части алгоритма	Примечания
Тонелли в поле F_p	1/2	—	$O((\log p)^2 M(\log_2 p))$	—
	—	$O(\log_2 p) M(\log_2 p)$		В предположении гипотезы Римана
Тонелли в поле $GF(p^n)$	1/2	—	$O(\log q)^2 M(GF(q))$	$M(GF(q)) = O(M_{GF(p)}(n) \times \times M(\log_2 p))$
Чиполлы в поле F_p	$(p-1)/(2p)$	—	$12l((p+1)/2) M(\log_2 p) \sim \sim 12M(\log_2 p)\log_2 p$	—
			$4M(\log_2 p)\log_2 p$	Для чисел специального вида
В поле характеристики $q \equiv 3 (\mathrm{mod} 4)$	—	—	$O((\log_2 r) M(r)) + r^{1+o(1)}$; $O((\log_2 r) M(r))$	В оптимальном нормальном базисе или с корнем бинома
В поле $GF(3^s)$, $s = 2^k r$			$O(M(2^k) M(r) + \log_2 r\, M(r)) + + kr^{1+o(1)}$; $O(M(2^k) M(r) + \log_2 r\, M(r))$	
В поле четного порядка			$O(M_{GF(2)}(n)) O(n)$; 0	С корнем трехчлена в нормальном базисе
В кольце Z_{p^n}, $p > 2$			Сложность алгоритма в кольце $Z_p + O(nM(\log_2 p) M_p(n)$	$M_p(n)$ — сложность умножения n-разрядных чисел в p-ичной системе счисления; $M(\log_2 p)$ — сложность умножения в двоичной системе чисел, меньших p
В кольце Z_{2^n}			$O(n \cdot M(n))$	—

конечных полях характеристики два // Дискретная математика. 2013. № 1 (25). С. 3—32.

15. **Bernstein D.J.** Batch Binary Edwards // Proc. Crypto — 2009. 2009. Pp. 317—336.

16. **Bach E.** A Note to Square Roots in Finite Fields // IEEE Trans. Inform. Theory. 1990. No. 36. Pp. 1494—1498.

17. **Гашков С.Б., Сергеев И.С.** О сложности и глубине булевых схем для умножения и инвертирования в некоторых полях // Вестник МГУ. Серия «Математика. Механика». 2009. № 4. С. 3—7.

18. **Гашков С.Б., Сергеев И.С.** О применении метода аддиттивных цепочек для инвертирования в конечных полях // Дискретная математика. 2006. № 4 (18). С. 56—72.

19. **Гашков С.Б., Сергеев И.С.** Сложность вычислений в конечных полях // Фундаментальная и прикладная математика. 2012. № 4 (17). С. 95—131.

20. **Umans C.** Fast Polynomial Factorization and Modular Composition in Small Characteristic // Proc. 40th Symp. Theory of Computing. 2008. Pp. 481—490.

21. **Ahmadi O., Hankerson D., Menezes A.** Formulas for Cube Roots // Discrete Appl. Math. 2007. V. 155 (3). Pp. 260—270.

References

1. **Koblits N.** Kurs Teorii Chisel i Kriptografii. M.: Nauch. Izd-vo NTP, 2001. (in Russian).

2. **Vasilenko O.N.** Teoretiko-chislovye Algoritmy v Kriptografii. Izd-vo MTSNMO, 2003. (in Russian).

3. **Glukhov M.M., Kruglov I.A., Pichkur A.B., Cheremushkin A.V.** Vvedenie v Teoretiko-chislovye Metody Kriptografii. SPb. — M. — Krasnodar: Izd-vo Lan', 2011. (in Russian).

4. **Bolotov A.A., Gashkov S.B., Frolov A.B. Chasovskikh A.A.** Elementarnoe Vvedenie v Ellipticheskuyu Kriptografiyu. Kn. 1. Algebraicheskie i Algoritmicheskie Osnovy. M.: Izd-vo Lenand, 2019. (in Russian).

5. **Kredall R., Pomerans K.** Prostye Chisla. Kriptograficheskie i Vychislitel'nye Aspekty. M.: Izd-vo «Knizhnyy Dom «Liberkom», 2011. (in Russian).

6. **Venbo Mao.** Sovremennaya Kriptografiya. M.: Izdat. Dom Vil'yams, 2005. (in Russian).

7. **Furer M.** Faster Integer Multiplication. SIAM J. Computing. 2009;3;39:979—1005.

8. **Saha A., C., Kurur P., Saptharishi R.** Fast Integer Multiplication Using Modular Arithmetic. SIAM J. Computing. 2013;2;42:685—699.

9. **Harvey D., van der Hoeven J., Lecerf G.** Even Faster Integer Multiplication. J. Complexity. 2016;36: 1—30.

10. **Covanov S., Thome E.** Fast Integer Multiplication Using Generalized Fermat Primes. J. Mathematics of Computation. 2016;1:1—27.

11. **Bernstein D.J. Chuengsatiansup C., Lange T.** Curve41417:Karatsuba Revisited. Proc. Cryptographic Hardware and Embedded Systems. 2014:316—334.

12. **Bach E.** Explicit Bounds for Primality Testing and Related Problems. Math.Comp.1989;55:355—380.

13. **Harvey D.,van der Hoeven J., Lecerf G.** Faster Polynomial Multiplication Over Finite Fields. ArXive. arXiv:1407.3361. 2014.

14. **Gashkov S.B., Sergeev I.S.** O Slozhnosti i Glubine Bulevykh Skhem dlya Umnozheniya i Invertirovaniya v Konechnykh Polyakh Kharakteristiki Dva. Diskretnaya Matematika. 2013;1 (25):3—32. (in Russian).

15. **Bernstein D.J.** Batch Binary Edwards. Proc. Crypto — 2009. 2009:317—336.

16. **Bach E.** A Note to Square Roots in Finite Fields. IEEE Trans. Inform. Theory. 1990;36:1494—1498.

17. **Gashkov S.B., Sergeev I.S.** O Slozhnosti i Glubine Bulevykh Skhem dlya Umnozheniya i Invertirovaniya v Nekotorykh Polyakh. Vestnik MGU. Seriya «Matematika. Mekhanika». 2009;4:3—7. (in Russian).

18. **Gashkov S.B., Sergeev I.S.** O Primenenii Meto-da Addittivnykh Tsepochek dlya Invertirovaniya v Konechnykh Polyakh. Diskretnaya Matematika. 2006;4 (18): 56—72. (in Russian).

19. **Gashkov S.B., Sergeev I.S.** Slozhnost' Vychisleniy v Konechnykh Polyakh. Fundamental'naya i Prikladnaya Matematika. 2012;4 (17):95—131. (in Russian).

20. **Umans C.** Fast Polynomial Factorization and Modular Composition in Small Characteristic. Proc. 40th Symp. Theory of Computing. 2008:481—490.

21. **Ahmadi O., Hankerson D., Menezes A.** Formulas for Cube Roots. Discrete Appl. Math. 2007;155 (3): 260—270.

Сведения об авторах

Гашков Сергей Борисович — доктор физико-математических наук, профессор кафедры дискретной математики МГУ им. М.В. Ломоносова, e-mail: sbgashkov@gmail.com

Фролов Александр Борисович — доктор технических наук, профессор кафедры математического моделирования НИУ «МЭИ», e-mail: abfrolov@mail.ru

Попова Елизавета Петровна — студентка института автоматики и вычислительной техники НИУ «МЭИ», e-mail: popova.elizaveta@list.ru

Information about authors

Gashkov Sergey B. — Dr.Sci. (Phys.-Math.), Professor of Discrete Mathematics Dept., Lomonosov Moscow State University, e-mail: sbgashkov@gmail.com

Frolov Aleksandr B. — Dr.Sci. (Techn.), Professor of Mathematical Modeling Dept., NRU MPEI, e-mail: abfrolov@mail.ru

Popova Elizaveta P. — Student of Institute of Automatics and Computer Engineering, NRU MPEI, e-mail: popova.elizaveta@list.ru

Статья поступила в редакцию 09.11.2017