Глава 8

Модулярна аритметика

8.1 Извличане на квадратен корен по модул p^k, p нечетно просто.

Да разгледаме първо случая $k = 1$, т.е. решаването на сравнението

$$x^2 \equiv a \pmod{p}, \quad (a, p) = 1. \tag{8.1}$$

(Случаят $(a, p) = p$ е тривиален.)

Това сравнение може да има само прости корени и следователно всяко от двете му решения се повдига с описания в $ 3.3 метод (Hensel lifting) по единствен начин до корен по модул p^k. Следователно съществената част е решаването на сравнение (8.1), което ще опишем по-долу.

Съгласно критерия на Ойлер е необходимото и достатъчно условие (8.1) да има решение е

$$a^{\frac{p-1}{2}} \equiv 1 \pmod{p}. \tag{8.2}$$

Затова считаме, че (8.2) е изпълнено.

Твърдение 8.1.1 *Ако $p \equiv 3$ или $p \equiv 7$ по модул 8, то*

$$x \equiv \pm a^{\frac{p+1}{4}} \pmod{p}$$

са двете решения на (1.1).

Доказателство. Използвайки (1.2) получаваме

$$x^2 = a^{\frac{p+1}{2}} = a^{\frac{p-1}{2}}.a \equiv a \pmod{p}.$$

Твърдение 8.1.2 *Ако $p \equiv 5 \pmod 8$, то*

$$x \equiv \begin{cases} \pm a^{\frac{p+3}{8}}, & \text{при } a^{\frac{p-1}{4}} \equiv 1 \pmod{p} \\ \pm 2^{\frac{p-1}{4}} a^{\frac{p+3}{8}}, & \text{при } a^{\frac{p-1}{4}} \equiv -1 \pmod{p} \end{cases}$$

са двете решения на (1.1).

Доказателство. В първия случай имаме

$$x^2 = a^{\frac{p+3}{4}} = a.a^{\frac{p-1}{4}} \equiv a \pmod{p}.$$

Във втория случай тъй като $\left(\frac{2}{p}\right) = -1$ за разглежданите стойности на p, то $2^{\frac{p-1}{2}} \equiv -1 \pmod{p}$. (Това показва, че 2 е примитивен корен по модул p и $2^{\frac{p-1}{4}} \equiv \sqrt{-1} \pmod{p}$.) Следователно

$$x^2 = 2^{\frac{p-1}{2}} a^{\frac{p-1}{4}}.a \equiv (-1)(-1)a \equiv a \pmod{p}.$$

Пример 8.1.1 Да се реши сравнението $x^2 \equiv 3 \pmod{37}$.
$p = 37 = 8.4 + 5$ Тогава

$$3^{\frac{p+3}{4}} = 3^{10} \equiv -3 \not\equiv 1 \pmod{37}.$$

Следователно

$$x \equiv \pm 2^9.3^5 \equiv \pm 15 \pmod{37}.$$

Упражнение 8.1.1 *Да се реши сравнението* $x^2 \equiv 6 \pmod{53}$.

Случаят $p \equiv 1 \pmod{8}$

В този случай не може да дадем формула за решението, но ще опишем алгоритъма на Tonelli и Shanks.

Нека $p - 1 = 2^e m$, където $m = 2t + 1$ е нечетно. Алгоритъмът се базира на факта, че Силовата подгрупа G_2, $|G_2| = 2^e$ има нечетен индекс $m = (\mathbb{Z}_p^* : G_2)$ и най-често е малка (e не е голямо).

1. Намираме пораждащ z на $G_2 =< z >$, т. е. $o(z) = 2^e$:
   Избираме $\gamma \in \mathbb{Z}_p^*$, което е к вадратичен неостатък, т. е. $\gamma^{\frac{p-1}{2}} \equiv -1 \pmod{p}$. Вероятността произволен елемент да е квадратичен неостатък е $1/2$, така че много бързо се намира подходящото γ. Тъй като

   $$(\gamma^m)^{2^{e-1}} = \gamma^{2^{e-1}m} = \gamma^{\frac{p-1}{2}} \equiv -1 \pmod{p},$$

   то $o(\gamma^m) = 2^e$. Следователно може да положим

   $$z = \gamma^m.$$

2. Намираме n, такова че $a^m = z^{2n}$.
   Тъй като $(a^m)^{2^{e-1}} = a^{\frac{p-1}{2}} \equiv 1 \pmod{p}$, то редът $o(a^m)$ е делител на 2^{e-1}. Следователно $a^m \in G_2^2 =< z^2 >$, т. е. $a^m = z^{2n}$. Ако e е малко, то най-бързо е да намерим предварително $< z^2 >$ и сравнявайки да определим n. В общия случай всеки алгоритъм за намиране на дискретен логаритъм върши работа.

3. Полагаме $x = z^n a^{-t}$
   Наистина

   $$x^2 = z^{2n} a^{-2t} \equiv a^m a^{-2t} = a \pmod{p}.$$

Пример 8.1.2 Да се реши сравнението $x^2 \equiv 2 \pmod{41}$.

$p = 41 = 8.5 + 1$ т.е. $m = 5,\ t = 2,\ e = 3$. Сравнението има решение, тъй като 2 е квадратичен остатък: $\left(\frac{2}{41}\right) = 1$. От $3^4 \equiv -1 \pmod{41}$ получаваме че $3^{20} \equiv -1 \pmod{41}$, т.е. 3 е квадратичен неостатък и полагаме $z = 3^5 \equiv -3 \pmod{41}$. Следователно

$$G_2^2 =< z^2 >= \{\pm 1, \pm 9\}.$$

Намираме $2^5 = 32 \equiv z^6 \pmod{41}$, откъдето $y = z^3 \equiv 14 \pmod{41}$. Следователно

$$x = \pm y a^{-2} = \pm 14.2^{-2} = \pm 7.21 \equiv \pm 7.2^{-1} \equiv \pm 24 \pmod{41}.$$

Ето и един вариант на алгоритъма, който съдържа частта за намиране на дискретния логаритъм.

Алгоритъм

1. $u := z,\ b := a^m,\ x := a^{\frac{m+1}{2}},\ k := e.$
2. Намери минималното естествено $0 \le \nu < k$, така че $b^{2^\nu} \equiv 1 \pmod{p}$.
3. while $\nu > 0$ do
   $c := u^{2^{k-\nu-1}},\ u := c^2,\ x := xc,\ b := bu,$
   $k := \nu$ and go to 2.
4. stop and return x.

При инициализирането $x^2 \equiv ba \pmod{p}$. След всяка стъпка това сравнение остава в сила за новите x и b, но редът на b намалява.

Пример 8.1.3 Да се реши сравнението $x^2 \equiv 2 \pmod{113}$.

$113 = 2^4.7 + 1$, т. е. $m = 7,\ e = 4$. Тъй като $\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}} = 1$, то 2 е квадратичен остатък, т. е. $2^{56} \equiv 1 \pmod{113}$.

Намираме квадратичен неостатък по модул 113:

$$\left(\frac{3}{113}\right) = (-1)^{56.1}\left(\frac{113}{3}\right) = \left(\frac{-1}{3}\right) = -1.$$

Полагаме $\gamma = 3$ и $z = 3^7 \equiv 40 \pmod{113}$

1. $u := 40,\ b := 2^7 \equiv 15 \pmod{113},\ x := 2^4 = 16,\ k := e = 4.$
2. $b^2 = 225 \equiv -1$, i.e. $b^{2^2} \equiv 1 \pmod{113}$. Следователно $\nu = 2$.
3. $\nu = 2 > 0$ и следователно $c := u^{2^{4-2-1}} = u^2 = 40^2 \equiv 18 \pmod{113}$.

   $x := xc = 16.18 \equiv 62 \pmod{113}$
   $u := c^2 = 18^2 \equiv -15 \pmod{113}$
   $b := bu = 15(-15) = -225 \equiv 1 \pmod{113}$.

$2'$. $b \equiv 1$, т.е. $\nu = 0$. go to 4.

4. Търсеното решение е
   $x \equiv \pm 62 \pmod{113}$

Упражнение 8.1.2 *Да се реши сравнението* $x^2 \equiv 5 \pmod{40961}$.

($p = 2^{13}.5 + 1,\quad m = 5,\ e = 13,\ \gamma = 3,\ z = 243.$)

8.1.1 Представяне на числата във вида $x^2 + dy^2$.

Твърдение 8.1.3 *Ако $p = x^2 + dy^2$, то $-d$ е квадратичен остатък по модул p, т. е. $\left(\frac{-d}{p}\right) = 1$.*

Доказателството е очевидно: $(xy^{-1})^2 \equiv -d \pmod{p}$.

Алгоритъм 1 ***(Cornacchia)***

1. Проверяваме дали $\left(\frac{-d}{p}\right) = 1$. Ако не е изпълнено: отказ.

2. Решаваме $z^2 \equiv -d \pmod{p}$. От двете решения $\pm z$ което е $> p/2$ го означаваме с a, другото (което е $< p/2$) с b. Докато $b > \sqrt{p}$ прилагаме алгоритъма на Евклид за двойката (a, b), т. е. $(a, b) := (b,\ a \pmod{b})$.

3. Ако $p - b^2 = dt^2$, то полагаме $x := b, \quad y = t$. В противния случай: отказ (представянето е невъзможно).

Пример 8.1.4 Да се представи $p = 31$ във вида $x^2 + 3y^2$. Със свойствата на символа на Льожандър и квадратичния закон за реципрочност се проверява, че -3 е квадратичен остатък по модул 31. $(\pm 11)^2 \equiv -3 \pmod{31}$. Тъй като $-11 \equiv 20 \pmod{31}$ последователно получаваме двойките $(20, 9),\ (9, 2)$. Числото $2 < \sqrt{31}$.
Сега намираме $31 - 2^2 = 27 = 3.3^2$, което дава $31 = 2^2 + 3.3^2$.

Пример 8.1.5 Да се представи $p = 43$ във вида $x^2 + 5y^2$. Със свойствата на символа на Льожандър и квадратичния закон за реципрочност се проверява, че -5 е квадрат по модул 43. Лесно се намира и решението $(\pm 9)^2 \equiv -5 \pmod{43}$. Тъй като $-9 \equiv 34 \pmod{43}$ последователно получаваме двойките $(34, 9), \quad (9, 7), \quad (7, 2)$. Числото $2 < \sqrt{43}$. $43 - 2^2 = 39$, което не се дели на 5. Следователно представянето е невъзможно.

8.2 Метод на Монгомери за умножение и повдигане в степен.

Методът на Монгомери е метод за аритметика по голям нечетен модул $m,\ (m, 2) = 1$. Нужда от такива изчисления възниква при реализацията на редица криптографски протоколи.

При реализацията на метода на Монгомери числата се считат за записани в B-ична бройна система, където $B = 2^k$, където k най-често е дължината на използваната дума в компютъра. Едно число се превръща от двоичен запис в B-ичен като битовете му се групират по k, т.е. $A = a_{n-1}B^{n-1} + a_{n-2}B^{n-2} + \cdots + a_1B + a_0,\ 0 \le a_i \le 2^k - 1$ е запис в B-ична бройна система, а всяка "цифра"a_i се определя с k бита запис. Като двоично числото A има дължина kn бита. Следователно, ако искаме простия модул p да е с дължина $L = 1024$ бита, то трябва да обработваме B-ични числа с дължина $n = 64$ и да съхранявяме евентуално с дължина 128, т.е. 256 байта. Описаният по-долу метод на Монтгомери позволява умножението и повдигането в степен да се извършват чрез

умножение на едноразрядно число с многоразрядно, т.е. оперира се с $n+1$ разрядни числа (а не с $2n$).

Умножението и делението на числото A с B представляват преместване на двоичния му запис на k позиции, съответно, на ляво и на дясно. При преместване на дясно "изпадналите"k бита задават остатъка от делението, а полученото число е частното.

Функция msm(a,x): Умножава едноцифрено с многоцифрено B-ично число, т.е. $0 \le a \le B-1$, $X = x_{n-1}B^{n-1} + x_{n-2}B^{n-2} + \cdots + x_1B + x_0 = (x_{n-1}x_{n-2}\dots x_1x_0)_B < B^n$.

INPUT: a, x.
Variable: $A = (A_1A_0)_B$ двуцифрено B-ично число, т.е. 32 бита
OUTPUT: $y = ax = (y_ny_{n-1}\dots y_0)_B$

1. $A := 0$
2. For i from 0 to $n-1$ do
   $A := ax_i + A_1$, $y_i := A_0$
3. $y_n := A_1$.

Изпълнението на функцията изисква n умножения (базисни умножения) на едноцифрени (B-ични) числа и още толкова събирания с евентуален пренос.

Сега да пристъпим към описанието на метода. Нека $R = B^n$, $m < R$ и $\gcd(m,R) = 1$. (Числото n обикновено е и избрано да е в сила и $B^{n-1} < m$.) Да означим с $m_1 \equiv -m^{-1} \pmod B$, $m_1 < B$.

Функция mng(x,y): Дава в резултат произведението $XYR^{-1} \pmod m$, наричано произведение на Монтгомери, където $X < m$, $Y < m$.

INPUT: $X = (x_{n-1}x_{n-2}\dots x_1x_0)_B$, $Y = (y_{n-1}\dots y_0)_B$.
Variable: $A = (a_n\dots a_1a_0)_B$ – $n+1$-цифрено B-ично число,
OUTPUT: $XYR^{-1}(\mod m)$.

1. $A := 0$
2. For i from 0 to $n-1$ do
   $u := \mathbf{msm}(a_0 + x_iy_0,\, m_1) \pmod B$
   $A := A + \mathbf{msm}(x_i, Y) + \mathbf{msm}(u, m)$
   $A := A/B$, т.е. преместваме A с k бита надясно. (изпадащите k бита трябва да са нули)
3. Ако $A > m$, то $A := A - m$.
4. $\mathbf{mng}(x,y) := A$.

Изпълнението на функцията изисква $2n$ базисни умножения, $2n$ изпълнения на функцията **msn** и $2n(n+1)+n$ събирания. Като вземем предвид казаното по-горе следва, че **mng** изисква $2n^2+2n$ базисни умножения и $4n^2+3n$ базисни събирания.

Да отбележим, че на всяка стъпка $A < 2m-1$. Наистина началната стойност е $0 < 2m-1$. Предполагаме, че на $i-1$-та стъпка неравенството остава в сила. Тогава за i-тата стъпка имаме

$$A < [(2m-1) + (B-1)(m-1) + (B-1)m]/B = (2mB - B + 1)/B < 2m-1.$$

Следователно A наистина е $n+1$-значно B-ично число.

Пример 8.2.1 Нека $m = 63 = 333_4$ и $B = 4$. Да изберем $R = 4^3 = 64$, т. е. $n = 3$. Нека $X = 26_{10} = 122_4$; $Y = 31_{10} = 133_4$. За произведението им по Монгомери получаваме

$$\mathbf{mng}(X, Y) = 26.31.64^{-1} \equiv 26.31.1 \equiv 50_{10} = 302_4 \pmod{63}.$$

Сега да проверим, че същият резултат се получава и с алторитъма.

$-m^{-1} = 63^{-1} \equiv -1(-1)^{-1} = 1 \pmod 4$, т. е. $m_1 = 1$.

1. Полагаме $A = 0$.
2. а) $i = 0, \quad a_0 = 0. \qquad u = (0 + 2.3).1 = 6 = 12_4 = 2 \pmod 4$.
   $A = 0 + 2.133_4 + 2.333_4 = 2330_4, \quad A = A/4 = 233_4$.
   б) $i = 1, \quad a_0 = 3. \qquad u = (3 + 2.3).1 = 9 = 21_4 = 1 \pmod 4$.
   $A = 233_4 + 2.133_4 + 1.333_4 = 233_4 + 332_4 + 333_4 = 2230_4, \quad A = A/4 = 223_4$.
   в) $i = 2, \quad a_0 = 3. \qquad u = (3 + 1.3).1 = 12_4 = 2 \pmod 4$.
   $A = 223_4 + 1.133_4 + 2.333_4 = 233_4 + 133_4 + 1332_4 = 3020_4, \quad A = A/4 = 302_4$.

$\mathbf{mng}(X, Y) = 302_4 = 50_{10}$

Функция exp(X, e): Пресмята X^e по модул m. Ползва като фиксирани параметри $R \pmod m$ и $R_1 = R^2 \pmod m < m$

INPUT: $X = (x_{n-1}x_{n-2}\dots x_1x_0)_B$, $e = (e_te_{t-1}\dots e_0)_B$.
Variable: A, $n+1$-цифрено B-ично число,
OUTPUT: $X^e \pmod m$.

1. $A := R \pmod m$
2. $X_1 := \mathbf{mng}(X, R_1)$, (което е равно на $XR \pmod m$)
3. For i from t down to 0 do
   $A := \mathrm{mng}(A, A)$ $(= A^2R^{-1} \pmod m)$
   ако $e_i = 1$, то $A := \mathrm{mng}(A, X_1)$ $(= AX \pmod m)$
4. $A := \mathbf{mng}(A, 1)$ $(= AR^{-1})$
5. $\mathbf{exp}(X, e) := A$

Изпълнението на функцията изисква най-много $2t + 2$ изпълнения на **mng**, което влече $4n(n + 1)(t + 1)$ базисни умножения и $(t + 1)n(4n + 3)$ базисни събирания.

Пример 8.2.2 *Да пресметнем* 13^5 *по модул* m= 61.

Избираме $R = 64 \equiv 3 \pmod{61}$ и $B = 4$. Изчисляваме $R^2 \equiv R_1 = 9 \pmod{61}$. Полагаме $X = 13_{10} = 31_4, \quad e = 5_{10} = 101_2$, т.е. $e_2 = 1,\ e_1 = 0,\ e_0 = 1$.

1. $A := R \equiv 3 \pmod{61}$.
2. $X_1 := \mathbf{mng}(13, 9) = 13.9.3^{-1} \equiv 39 \pmod{61}$.
3. $i = 2$
   $A := \mathrm{mng}(3, 3) = 3.3.3^{-1} \equiv 3 \pmod{61}$
   $e_2 = 1 \ \Rightarrow \ A := \mathrm{mng}(3, 39) = 3.39.3^{-1} \equiv 39 \pmod{61}$
   $i = 1$
   $A := \mathrm{mng}(39, 39) = 39.39.3^{-1} \equiv 19 \pmod{61}$
   $e_1 = 0 \ \Rightarrow \ A = 19$
   $i = 0$
   $A := \mathrm{mng}(19, 19) = 19.19.3^{-1} \equiv 39 \pmod{61}$
   $e_0 = 1 \ \Rightarrow \ A := \mathrm{mng}(39, 39) = 39.39.3^{-1} \equiv 19 \pmod{61}$
4. $A :== \mathrm{mng}(19, 1) = 19.1.3^{-1} \equiv 47 \pmod{61}$.

Следователно $13^5 \equiv 47 \pmod{61}$.