ALGORYTM FLOYDA ZNAJDOWANIA CYKLI

Dany jest ciąg

$$a_1, a_2, a_3, a_4, \ldots, a_n, \ldots$$

Zakładamy, że ten ciąg jest od pewnego miejsca okresowy:

$$a_{l+i+m} = a_{l+i}.$$

Zatem na początku mamy „ogon” długości l

$$a_1, a_2, \ldots, a_l,$$

po którym występuje powtarzający się cykl długości m

$$a_{l+1}, a_{l+2}, \ldots, a_{l+m}.$$

Nie znamy wartości l i m. Chcemy znaleźć indeksy $i \neq j$ takie, że

$$x_i = x_j.$$

Algorytm:
(1) $x := a_1$;
(2) $y := a_2$;
(3) $i := 1$;
(4) **repeat**
(5) $\quad i := i + 1$;
(6) $\quad x := \mathrm{next}(x)$;
(7) $\quad y := \mathrm{next}(\mathrm{next}(y))$
(8) **until** $x = y$;
(9) **return** $k = i$

Zauważmy, że niezmiennikiem pętli jest

$$x = a_i \quad \text{oraz} \quad y = a_{2i}.$$

Ćwiczenie. Najmniejszym indeksem k, dla którego

$$a_k = a_{2k}$$

jest

$$k = m \cdot \left(1 + \left\lfloor \frac{l}{m} \right\rfloor\right).$$

Przykład. Dany jest ciąg

$$a, b, c, d, e, f, g, h, i, j, h, i, j, h, i, j, h, i, j, h, i, j, \ldots,$$

w którym

$$l = 7 \quad \text{oraz} \quad m = 3.$$

Przebieg algorytmu:

i	x	y
1	a	b
2	b	d
3	c	f
4	d	h
5	e	j
6	f	i
7	g	h
8	h	j
9	i	i

Zauważmy, że

$$k = m \cdot \left(1 + \left\lfloor \frac{l}{m} \right\rfloor\right) = 3 \cdot (1+2) = 9.$$

ALGORYTM ρ POLLARDA

Jest to algorytm probabilistyczny znajdowania logarytmu dyskretnego. Ma on ten sam oczekiwany czas działania co algorytm „babystep giant-step", ale wymaga bardzo mało pamięci.

Zakładamy dla uproszczenia, że szukamy logarytmu dyskretnego

$$x = \log_\alpha \beta$$

w grupie G rzędu n, przy czym liczba n jest pierwsza.

Grupę G dzielimy na trzy podzbiory

$$G = S_1 \cup S_2 \cup S_3$$

podobnej wielkości. Kryterium podziału powinno być łatwo sprawdzalne. Tego podziału dokonujemy tak, by $1 \notin S_2$. Definiujemy ciąg elementów grupy G w następujący sposób:

$$x_0 = 1$$

oraz

$$x_{i+1} = \operatorname{next}(x_i) = \begin{cases} \beta \cdot x_i, & \text{jeśli} \quad x_i \in S_1, \\ x_i^2, & \text{jeśli} \quad x_i \in S_2, \\ \alpha \cdot x_i, & \text{jeśli} \quad x_i \in S_3, \end{cases}$$

Ten ciąg elementów grupy G definiuje dwa ciągi liczb naturalnych

$$a_0, a_1, a_2, \ldots \quad \text{oraz} \quad b_0, b_1, b_2, \ldots$$

takie, że

$$x_i = \alpha^{a_i} \cdot \beta^{b_i}.$$

Ciągi te można zdefiniować w następujący sposób:

$$a_0 = 0, \quad b_0 = 0$$

oraz

$$a_{i+1} = \begin{cases} a_i, & \text{jeśli} \quad x_i \in S_1, \\ 2a_i \,\mathbf{mod}\, n, & \text{jeśli} \quad x_i \in S_2, \\ (a_i+1) \,\mathbf{mod}\, n, & \text{jeśli} \quad x_i \in S_3, \end{cases}$$

$$b_{i+1} = \begin{cases} (b_i+1) \,\mathbf{mod}\, n, & \text{jeśli} \quad x_i \in S_1, \\ 2b_i \,\mathbf{mod}\, n, & \text{jeśli} \quad x_i \in S_2, \\ b_i, & \text{jeśli} \quad x_i \in S_3. \end{cases}$$

Teraz za pomocą algorytmu Floyda znajdujemy indeks k taki, że

$$x_k = x_{2k}.$$

Wtedy

$$\alpha^{a_k}\beta^{b_k} = \alpha^{a_{2k}}\beta^{b_{2k}},$$

czyli

$$\beta^{b_k - b_{2k}} = \alpha^{a_{2k} - a_k}.$$

Logarytmując obie strony przy podstawie α dostajemy

$$(b_k - b_{2k}) \cdot \log_\alpha \beta \equiv a_{2k} - a_k \pmod{n}.$$

Jeśli

$$b_k \not\equiv b_{2k} \pmod{n},$$

to stąd łatwo obliczamy $\log_\alpha \beta$.

Przykład. Element $\alpha = 2$ generuje w grupie $\mathbb{Z}_{383}$ podgrupę G rzędu $n = 191$. Niech $\beta = 228$. Szukamy zatem

$$\log_2 228$$

w grupie G.

Rozkładamy grupę G na trzy zbiory

$$S_1 \cup S_2 \cup S_3$$

w następujący sposób:

$$\begin{aligned} x \in S_1 \quad &\Leftrightarrow \quad x \equiv 1 \pmod 3, \\ x \in S_2 \quad &\Leftrightarrow \quad x \equiv 0 \pmod 3, \\ x \in S_3 \quad &\Leftrightarrow \quad x \equiv 2 \pmod 3. \end{aligned}$$

Obliczamy teraz kolejne wartości ciągów

$$x_i, \quad a_i, \quad b_i, \quad x_{2i}, \quad a_{2i}, \quad b_{2i}.$$

i	x_i	a_i	b_i	x_{2i}	a_{2i}	b_{2i}
1	228	0	1	279	0	2
2	279	0	2	184	1	4
3	92	0	4	14	1	6
4	184	1	4	256	2	7
5	205	1	5	304	3	8
6	14	1	6	121	6	18
7	28	2	6	144	12	38
8	256	2	7	235	48	152
9	152	2	8	72	48	154
10	304	3	8	14	96	118
11	372	3	9	256	97	119
12	121	6	18	304	98	120
13	12	6	19	121	5	51
14	144	12	38	144	10	104

Otrzymaliśmy

$$x_{14} = x_{28} = 144.$$

Teraz

$$\begin{gathered} r = b_{14} - b_{28} \equiv 38 - 104 \equiv 125 \pmod{191}, \\ r^{-1} \equiv 136 \pmod{191}, \\ s = a_{28} - a_{14} \equiv 10 - 12 \equiv 189 \pmod{191}, \\ r^{-1}s = 136 \cdot 189 \equiv 110 \pmod{191}. \end{gathered}$$

Zatem

$$\log_2 228 = 110.$$

ALGORYTM POHLIGA-HELLMANA

Znów obliczamy logarytm dyskretny w grupie G rzędu n. Zakładamy, że znamy rozkład n na czynniki pierwsze:

$$n = p_1^{\alpha_1} \cdot \ldots \cdot p_k^{\alpha_k}.$$

Chcemy obliczyć

$$x = \log_a b$$

w grupie G.

Zaczynamy od obliczeń wstępnych, wykonywanych tylko jeden raz dla grupy G. Obliczamy

$$r_{p_i,j} = a^{jn/p_i}$$

dla $i = 1, \ldots, k$ oraz $j = 0, \ldots, p_i - 1$.

Teraz dla każdej liczby pierwszej $p = p_i$ obliczamy

$$x \,\mathbf{mod}\, p^\alpha.$$

Niech więc

$$n = p^\alpha \cdot m,$$

gdzie

$$p \not| m.$$

Niech $y_0 = x$. Definiujemy ciągi liczb

$$\begin{array}{rcl} y_0 & = & x_0 + p \cdot y_1, \\ y_1 & = & x_1 + p \cdot y_2, \\ y_2 & = & x_2 + p \cdot y_3, \\ \ldots & & \ldots \\ y_{\alpha-1} & = & x_{\alpha-1} + p \cdot y_\alpha. \end{array}$$

Wtedy

$$x = x_0 + x_1 p + x_2 p^2 + \ldots + x_{\alpha-1} p^{\alpha-1} + y p^\alpha,$$

gdzie

$$0 \leq x_i < p \quad \text{oraz} \quad y \in \mathbb{Z}.$$

Obliczamy x_0:

$$b^{n/p} = a^{xn/p} = a^{y_0 n/p} = a^{(x_0 + p y_1) n/p} = a^{x_0 n/p} \cdot a^{y_1 n} = a^{x_0 n/p}.$$

Zatem

$$b^{n/p} = r_{p,x_0}$$

i stąd znajdujemy x_0.

Obliczamy x_1. Najpierw kładziemy

$$b_1 = b \cdot a^{-x_0}.$$

Wtedy

$$b_1 = a^x \cdot a^{-x_0} = a^{x - x_0} = a^{y_0 - x_0} = a^{p y_1}.$$

Następnie:

$$b_1^{n/p^2} = a^{py_1 n/p^2} = a^{y_1 n/p} = a^{(x_1+py_2)n/p} = a^{x_1 n/p} \cdot a^{y_2 n} = a^{x_1 n/p}.$$

Zatem

$$b_1^{n/p^2} = r_{p,x_1}$$

i stąd znajdujemy x_1.

Obliczamy x_2. Teraz kładziemy

$$b_2 = b_1 \cdot a^{-x_1 p}.$$

Wtedy

$$b_2 = a^{py_1} \cdot a^{-x_1 p} = a^{p(y_1-x_1)} = a^{p^2 y_2}.$$

Następnie:

$$b2^{n/p^3} = a^{p^2 y_2 n/p^3} = a^{y_2 n/p} = a^{(x_2+py_3)n/p} = a^{x_2 n/p} \cdot a^{y_3 n} = a^{x_2 n/p}.$$

Zatem

$$b_2^{n/p^3} = r_{p,x_2}$$

i stąd znajdujemy x_2 i tak dalej...

Wreszcie, jeśli mamy obliczone

$$x \,\mathbf{mod}\, p_i^{\alpha_i}$$

dla $i = 1, \ldots, k$, to za pomocą chińskiego twierdzenia o resztach obliczamy x

Przykład. Obliczamy

$$x = \log_2 62$$

modulo 181 (tzn. w grupie multyplikatywnej ciała $\mathbb{Z}_{181}$).

Zaczynamy od obliczeń wstępnych. Rozkładamy rząd grupy na czynniki:

$$180 = 2^2 \cdot 3^2 \cdot 5.$$

Teraz obliczamy

$$r_{p_i,j} = 2^{180j/p_i} \,\mathbf{mod}\, 181$$

dla $p_i = 2, 3, 5$ oraz $j = 0, \ldots, p_i - 1$.

Dla $p = 2$ mamy

$$\begin{aligned} r_{2,0} &= 2^{90\cdot 0} \,\mathbf{mod}\, 181 = 1, \\ r_{2,1} &= 2^{90\cdot 1} \,\mathbf{mod}\, 181 = 180. \end{aligned}$$

Dla $p = 3$ mamy

$$\begin{aligned} r_{3,0} &= 2^{60\cdot 0} \,\mathbf{mod}\, 181 = 1, \\ r_{3,1} &= 2^{60\cdot 1} \,\mathbf{mod}\, 181 = 48, \\ r_{3,2} &= 2^{60\cdot 2} \,\mathbf{mod}\, 181 = 132. \end{aligned}$$

Dla $p = 5$ mamy

$$\begin{aligned}
r_{5,0} &= 2^{36\cdot 0} \,\mathbf{mod}\, 181 = 1,\\
r_{5,1} &= 2^{36\cdot 1} \,\mathbf{mod}\, 181 = 59,\\
r_{5,2} &= 2^{36\cdot 2} \,\mathbf{mod}\, 181 = 42,\\
r_{5,3} &= 2^{36\cdot 3} \,\mathbf{mod}\, 181 = 125,\\
r_{5,4} &= 2^{36\cdot 4} \,\mathbf{mod}\, 181 = 135.
\end{aligned}$$

Mamy

$$a = 2 \quad \text{oraz} \quad b = 62.$$

Najpierw obliczamy

$$x \,\mathbf{mod}\, 2^2.$$

Mamy $p = 2$ oraz

$$x = x_0 + x_1 \cdot 2 + y \cdot 2^2,$$

gdzie $0 \leq x_0, x_1 < 2$ oraz $y \in \mathbb{Z}$.

Obliczamy

$$b^{n/p} = 62^{90} \,\mathbf{mod}\, 181 = 1.$$

Zatem

$$x_0 = 0.$$

Teraz

$$b_1 = ba^{-x_0} = b = 62.$$

Obliczamy

$$b_1^{n/p^2} = 62^{45} \,\mathbf{mod}\, 181 = 1.$$

Zatem

$$x_1 = 0.$$

Stąd dostajemy

$$x \equiv 0 + 0 \cdot 2 = 0 \pmod{2^2}.$$

Następnie obliczamy

$$x \,\mathbf{mod}\, 3^2.$$

Mamy

$$x = x_0 + x_1 \cdot 3 + y \cdot 3^2,$$

gdzie $0 \leq x_0, x_1 < 3$ oraz $y \in \mathbb{Z}$. Teraz mamy $p = 3$.

Obliczamy

$$b^{n/p} = 62^{60} \,\mathbf{mod}\, 181 = 48.$$

Zatem

$$x_0 = 1.$$

Teraz

$$b_1 = ba^{-x_0} = 62 \cdot 2^{-1} = 31.$$

Obliczamy

$$b_1^{n/p^2} = 31^{20} \mathbin{\mathbf{mod}} 181 = 1.$$

Zatem

$$x_1 = 0.$$

Stąd dostajemy

$$x \equiv 1 + 0 \cdot 3 = 1 \pmod{3^2}.$$

Wreszcie obliczamy

$$x \mathbin{\mathbf{mod}} 5^1.$$

Mamy

$$x = x_0 + y \cdot 5,$$

gdzie $0 \leq x_0 < 5$ oraz $y \in \mathbb{Z}$. Teraz mamy $p = 5$.

Obliczamy

$$b^{n/p} = 62^{36} \mathbin{\mathbf{mod}} 181 = 1.$$

Zatem

$$x_0 = 0,$$

czyli

$$x \equiv 0 \pmod{5}.$$

Teraz stosujemy chińskie twierdzenie o resztach, by rozwiązać układ kongruencji

$$\begin{cases} x \equiv 0 \pmod{4}, \\ x \equiv 1 \pmod{9}, \\ x \equiv 0 \pmod{5}. \end{cases}$$

Otrzymujemy rozwiązanie $x = 100$, czyli

$$2^{100} \equiv 62 \pmod{181}$$

lub inaczej

$$\log_2 62 = 100$$

w ciele $\mathbb{Z}_{181}$.

Jeśli znany jest rozkład liczby n na czynniki pierwsze, to czas działania algorytmu Pohliga-Hellmana wymaga wykonania

$$O\Big(\sum_{i=1}^{k} \alpha_i(\log n + \sqrt{p_i})\Big)$$

działań w grupie G.

Przykład. Rozważmy grupę multyplikatywną $\mathbb{Z}_p$, gdzie p jest następującą 107-cyfrową liczbą pierwszą p:

$$227088231986781039743145181950291021585250524967592855$$
$$96453269189798311427475159776411276642277139650833937.$$

Liczba $p-1$ ma rozkład na małe czynniki pierwsze:

$$p-1 = 2^4 \cdot 104729^8 \cdot 224737^8 \cdot 350377^4.$$

Zatem obliczanie logarytmów dyskretnych w tej grupie za pomocą algorytmu Pohliga-Hellmana jest szczególnie proste.

INDEX-CALCULUS ALGORITHM

Jest to obecnie najlepszy znany algorytm obliczania logarytmów dyskretnych. Algorytm ten wymaga wyboru tzw. **bazy rozkładu** (factor base): podzbioru S grupy G o tej własności, że znaczący odsetek elementów grupy można w efektywny sposób przedstawić w postaci iloczynu elementów ze zbioru S.

Opis algorytmu, który podamy, jest niekompletny z dwóch powodów. Po pierwsze, nie precyzujemy sposobu wyboru bazy rozkładu. Ta baza nie może być zbyt mała (gdyż wtedy za mało elementów grupy G ma żądany rozkład i długo trwa poszukiwanie takich elementów) i nie może być zbyt duża (gdyż prowadzi to do zbyt dużych układów równań). Po drugie, nie precyzujemy, w jaki sposób otrzymujemy potrzebne nam rozkłady na iloczyn czynników ze zbioru S.

Znamy właściwe bazy rozkładu i sposoby generowania potrzebnych rozkładów w przypadku pewnych grup cyklicznych takich jak $\mathbb{Z}_p$ czy $GF(2^m)$.

W przypadku grupy $\mathbb{Z}_p$ oczekiwany czas działania tego algorytmu wynosi (w przypadku zastosowania tzw. sita ciał liczbowych – *number field sieve*)

$$O(\exp((c+o(1))((\log p)^{1/3}(\log\log p)^{2/3}))).$$

Algorytm.

Dane: generator α grupy cyklicznej G rzędu n i element β grupy G;

Wynik: logarytm dyskretny $x = \log_\alpha \beta$;

(1) *Wybór bazy rozkładu.* Wybieramy taki podzbiór

$$S = \{p_1, p_2, \ldots, p_t\}$$

grupy G, by znaczna część elementów grupy G dawała się przedstawić w postaci iloczynów elementów zbioru S;

(2) *Wybór kongruencji liniowych.*

(2.1) Wybieramy losowo liczbę k taką, że $0 \leq k \leq n-1$ i obliczamy α^k;

(2.2) Próbujemy przedstawić α^k w postaci iloczynu elementów ze zbioru S:

$$\alpha^k = p_1^{c_1} \cdot \ldots \cdot p_t^{c_t},$$

gdzie $c_i \geq 0$. Jeśli się to uda, to logarytmujemy obie strony, otrzymując kongruencję liniową

$$c_1 \log_\alpha p_1 + \ldots + c_t \log_\alpha p_t \equiv k \pmod{n}$$

z niewiadomymi

$$\log_\alpha p_1, \ldots, \log_\alpha p_t;$$

(2.3) Powtarzamy kroki (2.1) i (2.2) dotąd, aż uda się zebrać $t+c$ kongruencji liniowych, gdzie c jest niedużą liczbą całkowitą (np. $c = 10$) taką, że otrzymany układ kongruencji liniowych ma dokładnie jedno rozwiązanie (z dużym prawdopodobieństwem);

(3) *Rozwiązanie układu kongruencji.* Rozwiązujemy układ $t+c$ kongruencji liniowych z t niewiadomymi i otrzymujemy wartości

$$\log_\alpha p_1, \ldots, \log_\alpha p_t;$$

(4) *Obliczanie logarytmu.*

(4.1) Wybieramy losowo liczbę k taką, że $0 \leq k \leq n-1$ i obliczamy $\beta \cdot \alpha^k$;

(4.2) Próbujemy przedstawić $\beta \cdot \alpha^k$ w postaci iloczynu elementów zbioru S:

$$\beta \cdot \alpha^k = p_1^{d_1} \cdot \ldots \cdot p_t^{d_t},$$

gdzie $d_i \leq 0$. Jeśli to się nie uda, to powtarzamy krok (4.1). Jeśli się uda, to logarytmujemy obie strony otrzymanej równości:

$$x = \log_\alpha \beta = (d_1 \log_\alpha p_1 + \ldots + d_t \log_\alpha p_t - k) \,\mathbf{mod}\, n.$$

PRZYKŁAD

Niech $p = 229$. Element $\alpha = 6$ jest generatorem grupy multyplikatywnej ciała $\mathbb{Z}_{229}$ rzędu $n = 228$. Niech $\beta = 13$. Obliczamy zatem

$$\log_6 13$$

w ciele $\mathbb{Z}_{229}$.

(1) Wybieramy bazę rozkładu
$$S = \{2, 3, 5, 7, 11\}.$$

(2) Znajdujemy sześć następujących rozkładów (pomijamy tu próby nieudane):

$$\begin{aligned}
6^{100} \,\mathbf{mod}\, 229 &= 180 = 2^2 \cdot 3^2 \cdot 5 \\
6^{18} \,\mathbf{mod}\, 229 &= 176 = 2^4 \cdot 11 \\
6^{12} \,\mathbf{mod}\, 229 &= 165 = 3 \cdot 5 \cdot 11 \\
6^{62} \,\mathbf{mod}\, 229 &= 154 = 2 \cdot 7 \cdot 11 \\
6^{143} \,\mathbf{mod}\, 229 &= 198 = 2 \cdot 3^2 \cdot 11 \\
6^{206} \,\mathbf{mod}\, 229 &= 210 = 2 \cdot 3 \cdot 5 \cdot 7
\end{aligned}$$

Oznaczmy teraz

$$y_1 = \log_6 2, y_2 = \log_6 3, y_3 = \log_6 5, y_4 = \log_6 7, y_5 = \log_6 11.$$

Otrzymujemy wtedy układ sześciu kongruencji liniowych z pięcioma niewiadomymi:

$$\begin{cases}
100 \equiv 2y_1 + 2y_2 + y_3 \pmod{228} \\
18 \equiv 4y_1 + y_5 \pmod{228} \\
12 \equiv y_2 + y_3 + y_5 \pmod{228} \\
62 \equiv y_1 + y_4 + y_5 \pmod{228} \\
143 \equiv y_1 + 2y_2 + y_5 \pmod{228} \\
206 \equiv y_1 + y_2 + y_3 + y_4 \pmod{228}
\end{cases}$$

(3) Rozwiązujemy układ kongruencji. Rozwiązaniem jest:

$$y_1 = 21, \quad y_2 = 208, \quad y_3 = 98, \quad y_4 = 107, \quad y_5 = 162.$$

(4) Przypuśćmy, że wylosowaliśmy liczbę $k = 77$. Wtedy

$$\beta \cdot \alpha^k = 13 \cdot 6^{77} \,\mathbf{mod}\, 229 = 147 = 3 \cdot 7^2,$$

więc
$$x = \log_6 13 = (y_2 + 2y_4 - k) \,\mathbf{mod}\, 228 = 117.$$

Uwagi.

(1) Najbardziej czasochłonne jest generowanie kongruencji liniowych. Można to zrobić równolegle na wielu komputerach i przesyłać dane do centralnego komputera, który rozwiązuje układ kongruencji liniowych.

(2) Obliczone logarytmy dyskretne elementów bazy rozkładu mogą być wykorzystane do znalezienia wielu logarytmów dyskretnych w danej grupie. Wykonanie kroku 4 jest znacznie mniej czasochłonne.

(3) Ten algorytm nie stosuje się bezpośrednio do podgrup grupy multyplikatywnej $\mathbb{Z}_p$ (α nie musi być generatorem). Przypuśćmy, że liczba pierwsza q jest dzielnikiem $p-1$

i α jest generatorem grupy cyklicznej rzędu q. Niech $l = (p-1)/g$. Modyfikujemy algorytm: znajdujemy generator γ grupy multyplikatywnej ciała $\mathbb{Z}_p$ i logarytmy dyskretne

$$y = \log_\gamma \alpha \quad \text{oraz} \quad z = \log_\gamma \beta.$$

Wtedy

$$x = \log_\alpha \beta = (z/l) \cdot (y/l)^{-1} \,\mathbf{mod}\, q.$$

(4) Ta wersja problemu logarytmu dyskretnego jest podstawą algorytmu podpisu cyfrowego U. S. Government NIST Digital Signature Algorithm.

HIPOTEZA DIFFIEGO-HELLMANA

Definicja. Załóżmy, że dana jest liczba pierwsza p i generator α grupy multyplikatywnej ciała $\mathbb{Z}_p$. Załóżmy następnie, że dane są elementy

$$\alpha^a \,\mathbf{mod}\, p \quad \text{oraz} \quad \alpha^b \,\mathbf{mod}\, p$$

ciała $\mathbb{Z}_p$. **Problem Diffiego-Hellmana** polega na znalezieniu elementu

$$\alpha^{ab} \,\mathbf{mod}\, p$$

ciała $\mathbb{Z}_p$.

Problem ten można uogólnić na przypadek dowolnej grupy cyklicznej skończonej.

Problem Diffiego-Hellmana łatwo sprowadza się (wielomianowo) do problemu logarytmu dyskretnego. Otwartym zagadnieniem jest, czy problem Diffiego-Hellmana jest równoważny obliczeniowo problemowi logarytmu dyskretnego. Wiadomo, że jeśli liczba $p-1$ rozkłada się na małe czynniki pierwsze, to te problemy są równoważne.

Problem Diffiego-Hellmana jest podstawą bezpieczeństwa systemów wspólnego wyboru klucza szyfrowego przez dwóch użytkowników.

SYSTEM MASSEYA-OMURY

Ustalamy liczbę pierwszą p. Każdy z dwóch użytkowników systemu wybiera swoją liczbę e taką, że $\mathrm{NWD}(e, p-1) = 1$ i oblicza liczbę d taką, że

$$de \equiv 1 \pmod{p-1}.$$

Sposób przesyłania wiadomości x od A do B:

(1) A wysyła do B liczbę $y = x^{e_A} \operatorname{\mathbf{mod}} p$;

(2) B wysyła do A liczbę $z = y^{e_B} \operatorname{\mathbf{mod}} p$;

(3) A wysyła do B liczbę $t = z^{d_A} \operatorname{\mathbf{mod}} p$;

(4) B oblicza $s = t^{d_B} \operatorname{\mathbf{mod}} p$; wtedy $s = x$.

Oznaczmy:

$$\begin{array}{rclrcl} E_A(x) & = & x^{e_A} \operatorname{\mathbf{mod}} p, & D_A(x) & = & x^{d_A} \operatorname{\mathbf{mod}} p, \\ E_B(x) & = & x^{e_B} \operatorname{\mathbf{mod}} p, & D_B(x) & = & x^{d_B} \operatorname{\mathbf{mod}} p. \end{array}$$

Wtedy oczywiście

$$E_A(D_A(x)) = D_A(E_A(x)) = x,$$
$$E_B(D_B(x)) = D_B(E_B(x)) = x.$$

W dowodzie równości $s = x$ wykorzystujemy następującą własność przemienności

$$E_A(E_B(x)) = E_B(E_A(x)).$$

SYSTEM EL GAMALA

Ustalamy liczbę pierwszą p i generator α grupy multyplikatywnej ciała $\mathbb{Z}_p$. Każdy użytkownik systemu wybiera losowo liczbę a taką, że $0 < a < p-1$ (jest to jego tajny klucz) i publikuje swój klucz publiczny $\alpha^a \operatorname{\mathbf{mod}} p$.

Jeśli chcemy wysłać wiadomość x do osoby, której kluczem publicznym jest $\alpha^a \operatorname{\mathbf{mod}} p$, to wybieramy losowo liczbę k taką, że $0 < k < p-1$ i wysyłamy parę liczb

$$(\alpha^k \operatorname{\mathbf{mod}} p, x \cdot \alpha^{ak} \operatorname{\mathbf{mod}} p).$$

Zauważmy, że umiemy obliczyć

$$\alpha^{ak} \operatorname{\mathbf{mod}} p = (\alpha^a)^k \operatorname{\mathbf{mod}} p.$$

Adresat może odczytać wiadomość x: najpierw oblicza

$$\alpha^{ak} \operatorname{\mathbf{mod}} p = (\alpha^k)^a \operatorname{\mathbf{mod}} p,$$

a następnie dzieli (modulo p)

$$x \cdot \alpha^{ak} \operatorname{\mathbf{mod}} p$$

przez

$$\alpha^{ak} \operatorname{\mathbf{mod}} p.$$

Bezpieczeństwo tego systemu jest oparte na hipotezie Diffiego-Hellmana.

SUMY KWADRATÓW

Problem. Dana jest liczba naturalna n. Czy istnieją liczby naturalne x i y takie, że

$$n = x^2 + y^2?$$

Jest to bardzo stary problem. Pewne sformułowania w *Arytmetyce* Diofantosa sugerują związek z tym problemem. Prawdziwą odpowiedź podał w 1625 roku holenderski matematyk Albert Girard i nieco później Fermat. Problem ten zazwyczaj wiążemy z nazwiskiem Fermata. Pierwszy znany dowód pochodzi od Eulera, choć prawdopodobnie Fermat też znał dowód.

Twierdzenie 1. Jeśli

$$n \equiv 3 \pmod 4,$$

to takie liczby x i y nie istnieją.

Dowód. Wystarczy zauważyć, że

$$x^2 \,\mathbf{mod}\, 4 \in \{0, 1\}. \quad \spadesuit$$

Twierdzenie 2. Jeśli

$$n = x^2 + y^2$$

oraz p jest liczbą pierwszą taką, że

$$p \mid n \quad \text{i} \quad p = 4k + 3,$$

to $p \mid x$ oraz $p \mid y$.

Dowód. Oczywiście

$$x^2 + y^2 \equiv 0 \pmod p.$$

Przypuśćmy, że

$$x \not\equiv 0 \pmod p.$$

Niech z będzie elementem odwrotnym do x modulo p:

$$xz \equiv 1 \pmod p.$$

Wtedy

$$(xz)^2 + (yz)^2 \equiv 0 \pmod p,$$

czyli

$$(yz)^2 \equiv -1 \pmod p.$$

Ale

$$\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}} = (-1)^{2k+1} = -1,$$

więc -1 jest nieresztą kwadratową modulo p, sprzeczność.

Zatem
$$x \equiv 0 \pmod{p},$$
skąd dostajemy także
$$y \equiv 0 \pmod{p}. \quad \spadesuit$$

Twierdzenie 3. Jeśli
$$n = x^2 + y^2$$
oraz p jest liczbą pierwszą taką, że
$$p \mid n \ \text{ i } \ p = 4k + 3,$$
to istnieje liczba parzysta m taka, że
$$p^m \mid n \ \text{ oraz } \ p^{m+1} \not| \, n.$$

Dowód. Z poprzedniego twierdzenia wiemy, że
$$p \mid x \ \text{ oraz } \ p \mid y.$$
Zatem $p^2 \mid n$. Stąd wynika, że istnieją liczby x_1, y_1 i n_1 takie, że
$$x = px_1, \quad y = py_1 \ \text{ oraz } \ n = p^2 n_1.$$
Wtedy
$$p^2 x_1^2 + p^2 y_1^2 = p^2 n,$$
czyli
$$x_1^2 + y_1^2 = n_1.$$
Jeśli $p \mid n_1$, to powtarzamy rozumowanie. Istnieją x_2, y_2 i n_2 takie, że $n_2 < n$ oraz
$$x_2^2 + y_2^2 = n_2$$
i tak dalej. $\spadesuit$

GŁÓWNE TWIERDZENIE

Twierdzenie 4. Liczba naturalna n jest sumą dwóch kwadratów wtedy i tylko wtedy, gdy dla każdej liczby pierwszej p takiej, że $p \mid n$ oraz $p = 4k + 3$ istnieje liczba parzysta m taka, że
$$p^m \mid n \ \text{ oraz } \ p^{m+1} \not| \, n.$$

Dowód. Zauważmy najpierw, że jeśli
$$n = x^2 + y^2,$$

to każda liczba postaci m^2n też jest sumą dwóch kwadartów:

$$m^2n = (mx)^2 + (my)^2.$$

Zatem wystarczy udowodnić nasze twierdzenie w przypadku, gdy liczba n jest bezkwadratowa. Z założenia wynika, że wszystkie dzielniki pierwsze liczby n są wtedy postaci $4k + 1$.

Zauważmy następnie, że jeśli

$$n = x^2 + y^2 \quad \text{oraz} \quad m = t^2 + s^2,$$

to iloczyn liczb n i m też jest sumą dwóch kwadratów:

$$nm = (x^2 + y^2) \cdot (t^2 + s^2) = (xt + ys)^2 + (xs - yt)^2.$$

Oczywiście $2 = 1^2 + 1^2$. Zatem wystarczy dowieść, że każda liczba pierwsza p postaci $p = 4k + 1$ jest sumą dwóch kwadratów. Jest to tzw. twierdzenie Fermata, które za chwilę udowodnimy. ♠

TWIERDZENIE FERMATA

Twierdzenie 5. Jeśli liczba p jest pierwsza oraz $p = 4k + 1$, to istnieją liczby naturalne x i y takie, że

$$n = x^2 + y^2.$$

Dowód. (D. Zagier, 1989) Definiujemy następujący zbiór:

$$S = \{(x, y, z) \in \mathbb{N}^3 : \; x^2 + 4yz = p\}.$$

Zauważamy, że zbiór S jest niepusty:

$$(1, 1, k) \in S$$

i skończony (bo $x, y, z < \sqrt{p}$).

Zbiór S można podzielić na trzy podzbiory rozłączne:

$$\begin{aligned}
S_1 &= \{(x, y, z) \in S : \; x < y - z\}, \\
S_2 &= \{(x, y, z) \in S : \; y - z < x < 2y\}, \\
S_3 &= \{(x, y, z) \in S : \; 2y < x\}.
\end{aligned}$$

Łatwo pokazujemy, że

$$S = S_1 \cup S_2 \cup S_3.$$

Mianowicie:

$$x = y - z \quad \Rightarrow \quad (y+z)^2 = p$$

oraz

$$x = 2y \quad \Rightarrow \quad 4y(y+z) = p.$$

Teraz definiujemy funkcję $f : S \to S$ wzorami

$$f(x,y,z) = \begin{cases} (x+2z, z, y-x-z) & \text{jeśli } (x,y,z) \in S_1, \\ (2y-x, y, x-y+z) & \text{jeśli } (x,y,z) \in S_2, \\ (x-2y, x-y+z, y) & \text{jeśli } (x,y,z) \in S_3. \end{cases}$$

Można łatwo sprawdzić, że

$$\begin{aligned} (x,y,z) \in S_1 \quad &\Rightarrow \quad f(x,y,z) \in S_3, \\ (x,y,z) \in S_2 \quad &\Rightarrow \quad f(x,y,z) \in S_2, \\ (x,y,z) \in S_3 \quad &\Rightarrow \quad f(x,y,z) \in S_1. \end{aligned}$$

Następnie sprawdzamy, że funkcja f jest **inwolucją**, tzn.

$$f(f(x,y,z)) = (x,y,z).$$

Funkcja f ma punkt stały. Jest nim trójka $(1,1,k)$. Mianowicie wtedy

$$(1,1,k) \in S_2$$

oraz

$$f(1,1,k) = (2-1, 1, 1-1+k) = (1,1,k).$$

Jest to jedyny punkt stały funkcji f. Jeśli bowiem

$$f(x,y,z) = (x,y,z),$$

to $(x,y,z) \in S_2$ i wtedy

$$(2y-x, y, x-y+z) = (x,y,z).$$

Stąd wynika że $x = y$ i wtedy

$$y(y+4z) = p.$$

Zatem $x = y = 1$ i $z = k$.

Zbiór S ma zatem **nieparzystą** liczbę elementów. Stąd wynika, że każda inwolucja zbioru S ma punkt stały.

Rozważmy funkcję $g : S \to S$ określoną wzorem:

$$g(x,y,z) = (x,z,y).$$

Oczywiście funkcja g jest inwolucją, a więc ma punkt stały.

Niech trójka (x, y, z) będzie punktem stałym inwolucji g. Wtedy

$$(x, y, z) = g(x, y, z) = (x, z, y),$$

a więc $y = z$. Zatem

$$p = x^2 + 4yz = x^2 + 4y^2 = x^2 + (2y)^2,$$

co kończy dowód. ♠

DOWÓD EULERA

Dowód podany przez Zagiera jest nieefektywny: nie pokazuje żadnej metody znalezienia punktu stałego inwolucji g. Wspomniany wcześniej dowód Eulera jest efektywny: można z niego odczytać algorytm.

Lemat. Jeśli p jest liczbą pierwszą oraz $p = 4k + 1$, to istnieją liczby naturalne x, y i m takie, że

$$mp = x^2 + y^2 \quad \text{oraz} \quad 0 < m < p.$$

Dowód. Ponieważ $p = 4k + 1$, więc -1 jest resztą kwadratową modulo p. Zatem istnieje liczba naturalna a taka, że $0 < a < p$ oraz

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

Stąd wynika, że istnieje liczba naturalna m taka, że

$$a^2 + 1 = mp.$$

Teraz mamy

$$mp = a^2 + 1 \leq (p - 1)^2 + 1 < p^2,$$

a zatem $m < p$. Wystarczy teraz położyć $x = a$ i $y = 1$. ♠

Zauważmy, że liczby x, y i m możemy wyznaczyć za pomocą znanych algorytmów.

Z lematu wynika, że istnieje liczba całkowita dodatnia m taka, że $m < p$, dla której równanie

$$mp = x^2 + y^2$$

z niewiadomymi x i y ma rozwiązanie. Jeśli $m = 1$, to twierdzenie jest udowodnione (i można znaleźć liczby x i y).

Przypuśćmy, że $m > 1$. Znajdziemy liczbę całkowitą dodatnią $m_1 < m$ o tej własności, że równanie

$$m_1 p = x_1^2 + y_1^2$$

ma rozwiązanie (i to rozwiązanie będziemy mogli znaleźć efektywnie).

Znajdujemy liczby a i b spełniające następujące warunki:

(1) $a \equiv x \pmod{m}$;

(2) $-m/2 < a \leq m/2$;

(3) $b \equiv y \pmod{m}$;

(4) $-m/2 < b \leq m/2$.

Wtedy
$$a^2 + b^2 \equiv x^2 + y^2 = mp \equiv 0 \pmod{m}.$$
Zatem istnieje liczba naturalna m_1 taka, że
$$m_1 m = a^2 + b^2.$$

Zauważmy, że liczby a, b i m_1 wyznaczamy efektywnie.
Teraz mamy
$$(a^2 + b^2)(x^2 + y^2) = (m_1 m)(mp) = m_1 m^2 p.$$
Ale także
$$(a^2 + b^2)(x^2 + y^2) = (ax + by)^2 + (ay - bx)^2.$$
Stąd wynika, że
$$(ax + by)^2 + (ay - bx)^2 = m_1 m^2 p.$$
Ponadto
$$ax + by \equiv x^2 + y^2 \equiv 0 \pmod{m},$$
$$ay - bx \equiv xy - yx \equiv 0 \pmod{m}.$$

Zatem liczby
$$x_1 = \frac{ax + by}{m} \quad \text{oraz} \quad y_1 = \frac{ay - bx}{m}$$
są całkowite.

Teraz zauważamy, że
$$x_1^2 + y_1^2 = \frac{(ax + by)^2 + (ay - bx)^2}{m^2} = m_1 p.$$

Liczby x_1 i y_1 tworzą zatem rozwiązanie równania
$$m_1 p = x_1^2 + y_1^2.$$

Wystarczy więc pokazać, że
$$0 < m_1 < m.$$

Najpierw pokazujemy, że $m_1 \neq 0$. Gdyby bowiem $m_1 = 0$, to mielibyśmy $a^2 + b^2 = 0$, czyli $a = b = 0$. Wtedy
$$x \equiv y \equiv 0 \pmod{m},$$

czyli $m \mid x$ oraz $m \mid y$. Zatem

$$m^2 \mid x^2 + y^2.$$

Ale $x^2 + y^2 = mp$, więc

$$m^2 \mid mp,$$

czyli

$$m \mid p.$$

To jednak jest niemożliwe, gdyż $1 < m < p$.

Z drugiej strony,

$$|a| \le \frac{m}{2} \quad \text{oraz} \quad |b| \le \frac{m}{2}.$$

Zatem

$$m_1 m = a^2 + b^2 \le 2 \cdot \frac{m^2}{4} = \frac{m^2}{2},$$

skąd wynika, że

$$m_1 \le \frac{m}{2} < m.$$

PRZYKŁAD

Niech $p = 277$. Wtedy

$$277 = 4 \cdot 69 + 1,$$

a więc liczba p jest postaci $4k + 1$.

Najpierw rozwiązujemy kongruencję kwadratową

$$a^2 \equiv -1 \pmod{277}.$$

Rozwiązaniem jest $a = 60$.

Teraz obliczamy m. Mamy

$$\frac{60^2 + 1}{277} = \frac{3601}{277} = 13,$$

czyli

$$13 \cdot 277 = 60^2 + 1^2.$$

Zatem

$$x = 60, \quad y = 1, \quad m = 13.$$

Znajdujemy teraz liczby a i b.

$$60 \,\mathbf{mod}\, 13 = 8, \quad 1 \,\mathbf{mod}\, 13 = 1.$$

Ponieważ $8 > \frac{13}{2}$ oraz $1 < \frac{13}{2}$, więc

$$a = 8 - 13 = -5, \quad b = 1.$$

Teraz znajdujemy m_1:

$$\frac{a^2 + b^2}{m} = \frac{25 + 1}{13} = 2.$$

Zatem $m_1 = 2$.

Wreszcie obliczamy x_1 i y_1:

$$x_1 = \frac{ax + by}{m} = \frac{(-5) \cdot 60 + 1 \cdot 1}{13} = \frac{-299}{13} = -23$$

oraz

$$y_1 = \frac{ay - bx}{m} = (-5) \cdot 1 - 1 \cdot 60 = \frac{-65}{13} = -5.$$

Stąd otrzymujemy

$$2 \cdot 277 = (-23)^2 + (-5)^2 = 23^2 + 5^2.$$

Ponieważ $m_1 > 1$, więc powtarzamy rozumowanie. Mamy więc

$$x_1 = 23, \quad y_1 = 5, \quad m_1 = 2.$$

Obliczamy najpierw a i b.

$$23 \,\mathbf{mod}\, 2 = 1, \quad 5 \,\mathbf{mod}\, 2 = 1.$$

Zatem $a = 1$ i $b = 1$.

Teraz obliczamy m_2.

$$\frac{a^2 + b^2}{m_1} = \frac{1 + 1}{2} = 1.$$

Zatem $m_2 = 1$

Wreszcie obliczamy x_2 i y_2:

$$x_2 = \frac{ax_1 + by_1}{m_1} = \frac{1 \cdot 23 + 1 \cdot 5}{2} = \frac{28}{2} = 14$$

oraz

$$y_2 = \frac{ay_1 - bx_1}{m_1} = \frac{1 \cdot 5 - 1 \cdot 23}{2} = \frac{-18}{2} = -9.$$

A więc ostatecznie

$$277 = 14^2 + (-9)^2 = 14^2 + 9^2.$$

INNE ALGORYTMY

Znane są inne sposoby otrzymywania rozkładu liczby pierwszej postaci $p = 4k + 1$ na sumę dwóch kwadratów:

$$p = x^2 + y^2.$$

Legendre w 1808 roku podał wzory na x i y wykorzystujące rozwinięcie liczby $\sqrt{p}$ na ułamek łańcuchowy.

Gauss w 1825 roku podał następujące wzory:

$$x \equiv \frac{1}{2} \cdot \binom{2k}{k} \pmod{p}, \quad y \equiv x \cdot (2k)! \pmod{p},$$

przy czym x i y należy dobrać tak, by

$$|x|, |y| \leq \frac{p}{2}.$$

Serret w 1848 roku (i nieco później Hermitte) podał inne wzory wykorzystujące ułamki łańcuchowe. Wreszcie Jacobsthal w 1906 roku podał wzory wykorzystujące sumy symboli Legendre'a. Dowody tych wszystkich wzorów są nieco dłuższe.

Uwaga. Rozkład liczby pierwszej p na sumę dwóch kwadratów

$$p = x^2 + y^2$$

jest jednoznaczny z dokładnością do kolejności składników i znaku liczb x i y. Nietrudny dowód pozostawimy jako ćwiczenie.

ALGORYTM CORNACCHII

W 1908 roku Giuseppe Cornacchia podał bardzo prosty algorytm znajdowania rozkładu liczby piewszej $p = 4k + 1$ na sumę dwóch kwadratów. Idea tego algorytmu była zawarta już w pracy Serreta.

Algorytm ten składa się zasadniczo z dwóch kroków.

Dane: liczba pierwsza p i liczba k taka, że $p = 4k + 1$;

Wyniki: liczby naturalne x i y takie, że $p = x^2 + y^2$;

(1) Znajdujemy liczbę d taką, że $d^2 \equiv -1 \pmod{p}$;

(2) Wykonujemy algorytm Euklidesa dla liczb $r_0 = p$ i $r_1 = d$:

$$
\begin{array}{rcll}
r_0 & = & r_1 q_1 + r_2, & 0 < r_2 < r_1, \\
r_1 & = & r_2 q_2 + r_3, & 0 < r_3 < r_2, \\
r_2 & = & r_3 q_3 + r_4, & 0 < r_4 < r_3, \\
r_3 & = & r_4 q_4 + r_5, & 0 < r_5 < r_4, \\
\ldots & & \ldots & \ldots \\
r_i & = & r_{i+1} q_{i+1} + r_{i+2}, & 0 < r_{i+2} < r_{i+1}, \\
\ldots & & \ldots & \ldots \\
r_{n-4} & = & r_{n-3} q_{n-3} + r_{n-2}, & 0 < r_{n-2} < r_{n-3}, \\
r_{n-3} & = & r_{n-2} q_{n-2} + r_{n-1}, & 0 < r_{n-1} < r_{n-2}, \\
r_{n-2} & = & r_{n-1} q_{n-1} + r_n, & 0 < r_n < r_{n-1}, \\
r_{n-1} & = & r_n q_n + r_{n+1}, & r_{n+1} = 0.
\end{array}
$$

Niech

$$m = \min\{i : \ r_i < \sqrt{p}\}.$$

Wtedy kładziemy $x = r_i$ oraz $y = r_{i+1}$.

Krok pierwszy wykonujemy w następujący sposób. Po pierwsze znajdujemy nieresztę kwadratową m modulo p (za pomocą znanego algorytmu probabilistycznego).

Wtedy z kryterium Eulera wynika, że

$$m^{\frac{p-1}{2}} = m^{2k} = \left(\frac{m}{p}\right) = -1.$$

Niech

$$c = m^k \,\mathbf{mod}\, p.$$

Wtedy

$$c^2 \equiv m^{2k} \equiv -1 \pmod{p},$$

a więc c jest pierwiastkiem kwadratowym z -1 modulo p.

Drugim pierwiastkiem jest oczywiście $p - c$. Jako d wybieramy **mniejszy** z nich:

$$d = \min\{c, p - c\}.$$

Zauważmy też, że liczba n jest względnie pierwsza z p; stąd wynika, że liczba c też jest względnie pierwsza z p, a zatem również

$$\mathrm{NWD}(p, d) = 1.$$

W kroku drugim wykonujemy algorytm Euklidesa dla liczb p i d. Przyjmujemy więc

$$r_0 = p, \quad r_1 = d.$$

Dla kolejnych $i = 2, 3, \ldots, n+1$ obliczamy wyrazy ciągu r_i. Wiemy, że

$$r_n = \mathrm{NWD}(r_0, r_1).$$

Liczby p i d są względnie pierwsze, a więc

$$r_n = 1, \quad r_{n+1} = 0.$$

W czasie wykonywania algorytmu Euklidesa obliczamy również kolejne ilorazy q_i. Przyjmiemy, że

$$q_0 = q_{n+1} = 0.$$

Będziemy również obliczać wyrazy następującego ciągu t_i:

$$t_0 = 0, \quad t_1 = 1, \quad t_{i+2} = t_i - q_{i+1}t_{i+1}.$$

Udowodnimy teraz kilka prostych własności ciągu t_i. Najpierw pokażemy przez indukcję, że

$$t_i r_1 \equiv r_i \pmod{r_0}$$

dla $i = 0, 1, \ldots, n+1$.

Warunek początkowy dla $i = 0$ oraz $i = 1$ jest oczywisty. Załóżmy więc, że

$$t_i r_1 \equiv r_i \pmod{r_0}$$

oraz

$$t_{i+1} r_1 \equiv r_{i+1} \pmod{r_0}.$$

Udowodnimy, że

$$t_{i+2} r_1 \equiv r_{i+2} \pmod{r_0}.$$

Otóż

$$\begin{aligned} t_{i+2} r_1 &= (t_i - q_{i+1}t_{i+1})r_1 = t_i r_1 - q_{i+1}t_{i+1}r_1 \equiv \\ &\equiv r_i - q_{i+1}r_{i+1} = r_{i+2} \pmod{r_0}. \end{aligned}$$

Następnie zauważmy, że

$$t_{2i+1} > 0 \text{ oraz } t_{2i} < 0.$$

Ponadto ciąg

$$|t_0|, \quad |t_1|, \quad |t_2|, \quad |t_3|, \quad \ldots, \quad |t_{n+1}|$$

jest rosnący.

Wreszcie zauważmy, że

$$|t_{i+1}| = q_i|t_i| + |t_{i-1}| \text{ oraz } |t_{i-1}| < |t_i|.$$

Zatem ciąg

$$|t_{n+1}|, \quad |t_n|, \quad \ldots, \quad |t_1|, \quad |t_0|$$

jest otrzymanym za pomocą algorytmu Euklidesa ciągiem reszt dla danych początkowych $|t_{n+1}|$ i $|t_n|$.

PRZYKŁAD

Niech $p = 364289$. Wtedy
$$p = 4 \cdot 91072 + 1,$$
czyli $p = 4k + 1$ dla $k = 91072$. Liczba $m = 3$ jest nieresztą kwadratową modulo p. Następnie
$$c = m^k \,\mathbf{mod}\, p = 3^{91072} \,\mathbf{mod}\, 364289 = 202871.$$
Wtedy $p - c = 161418$. Ponieważ $p - c < c$, więc przyjmujemy
$$d = p - c = 161418.$$

Popatrzmy teraz na wartości ciągów r_i, q_i oraz t_i otrzymane za pomocą algorytmu Euklidesa:

i	r_i	q_i	t_i
0	364289	0	0
1	161418	2	1
2	41453	3	−2
3	37059	1	7
4	4394	8	−9
5	1907	2	79
6	580	3	−167
7	167	3	580
8	79	2	−1907
9	9	8	4394
10	7	1	−37059
11	2	3	41453
12	1	2	−161418
13	0	0	364289

Zaskakujące jest to, że ciąg
$$|t_{n+1}|, \quad |t_n|, \quad \ldots, \quad |t_1|, \quad |t_0|$$
jest dokładnie ciągiem reszt
$$r_0, \quad r_1, \quad ,\ldots, \quad r_{n+1},$$
wypisanym w odwrotnej kolejności. Nie jest to przypadek.

Okazuje się bowiem, że jeśli $p = 4k + 1$ i d jest mniejszym z dwóch pierwiastków z -1 modulo p, to ciągi r_i, q_i i t_i otrzymane za pomocą algorytmu Euklidesa mają następujące własności:

(1) Ciąg
$$|t_{n+1}|, \quad |t_n|, \quad \ldots, \quad |t_1|, \quad |t_0|$$
jest dokładnie ciągiem reszt
$$r_0, \quad r_1, \quad \ldots, \quad r_{n+1},$$

wypisanym w odwrotnej kolejności.

(2) Liczba n jest parzysta.

(3) Ciąg

$$q_0, \quad q_1, \quad q_2, \quad \ldots, \quad q_{\frac{n}{2}}, \quad q_{\frac{n}{2}+1}, \quad \ldots, \quad q_n, \quad q_{n+1}$$

jest symetryczny, tzn.

$$q_i = q_{n+1-i}.$$

(4) $p \mid r_i^2 + t_i^2$.

(5) $\frac{n}{2} = \min\{i : \ r_i < \sqrt{p}\}$.

Z tych pięciu własności wynika poprawność algorytmu Cornacchii.

Mianowicie z warunku (4) wynika, że

$$p \mid r_{\frac{n}{2}}^2 + t_{\frac{n}{2}}^2.$$

Z warunku (1) wynika, że

$$t_{\frac{n}{2}} = r_{\frac{n}{2}+1}.$$

Zatem

$$p \mid r_{\frac{n}{2}}^2 + r_{\frac{n}{2}+1}^2.$$

Z warunku 5 wynika, że

$$r_{\frac{n}{2}}, r_{\frac{n}{2}+1} < \sqrt{p},$$

skąd otrzymujemy

$$r_{\frac{n}{2}}^2 + r_{\frac{n}{2}+1}^2 < 2p.$$

Jedyną liczbą mniejszą od $2p$ podzielną przez p jest sama liczba p. Zatem

$$r_{\frac{n}{2}}^2 + r_{\frac{n}{2}+1}^2 = p.$$

Zauważmy, że wykonanie algorytmu możemy zakończyć wtedy, gdy znajdziemy pierwsze dwie reszty mniejsze od $\sqrt{p}$. Możemy to łatwo sprawdzić za pomocą warunku

$$r_i < |t_i|.$$

Wtedy bowiem

$$i = \frac{n}{2} + 1,$$

skąd wynika, że

$$t_i = t_{\frac{n}{2}+1},$$

czyli

$$r_i^2 + t_i^2 = r_{\frac{n}{2}+1}^2 + r_{\frac{n}{2}}^2 = p.$$

Zatem możemy przyjąć

$$x = r_i \quad \text{oraz} \quad y = t_i.$$

W naszym przykładzie $r_i < |t_i|$ po raz pierwszy dla $i = 7$. Wtedy

$$r_7 = 167 \quad \text{oraz} \quad t_7 = 580.$$

Zatem

$$364289 = r_7^2 + t_7^2 = 167^2 + 580^2.$$

WZÓR EULERA

Udowodnimy teraz własność (1). W tym celu zauważmy, że jeśli wykonamy algorytm Euklidesa dla liczb względnie pierwszych a i b, to otrzymany ciąg ilorazów

$$q_1, \quad q_2, \quad \ldots, \quad q_n$$

wyznacza jednoznacznie liczby a i b. Mamy bowiem

$$\begin{aligned}
r_n &= 1, \\
r_{n-1} &= r_n q_n = q_n, \\
r_{n-2} &= r_{n-1}q_{n-1} + r_n = q_n q_{n-1} + 1, \\
r_{n-3} &= r_{n-2}q_{n-2} + r_{n-1} = (q_n q_{n-1} + 1)q_{n-2} + q_n = \\
&= q_n q_{n-1} q_{n-2} + q_n + q_{n-2}, \\
r_{n-4} &= r_{n-3}q_{n-3} + r_{n-2} = \ldots = \\
&= q_n q_{n-1} q_{n-2} q_{n-3} + q_n q_{n-3} + q_{n-2} q_{n-3} + q_n q_{n-3} + 1
\end{aligned}$$

i tak dalej.

Można dość łatwo pokazać, że

$$a = f(q_1, q_2, \ldots, q_n),$$

gdzie wyrażenie $f(q_1, q_2, \ldots, q_n)$ jest sumą następujących iloczynów:

(1) $q_1 q_2 \cdot \ldots \cdot q_n$;

(2) iloczynów postaci $q_1 q_2 \cdot \ldots \cdot q_i q_{i+2} \cdot \ldots \cdot q_n$, tzn. iloczynów otrzymanych przez pominięcie w iloczynie wszystkich q_i dwóch kolejnych czynników;

(3) iloczynów postaci $q_1 q_2 \cdot \ldots \cdot q_i q_{i+2} \cdot \ldots \cdot q_j q_{j+2} \cdot \ldots \cdot q_n$, tzn. iloczynów otrzymanych przez pominięcie dwóch par kolejnych czynników i tak dalej.

PRZYKŁAD

Dla $n = 5$ otrzymamy

$$\begin{aligned}
f(q_1, q_2, q_3, q_4, q_5) &= q_1 q_2 q_3 q_4 q_5 + \\
&\quad + q_3 q_4 q_5 + q_1 q_4 q_5 + q_1 q_2 q_5 + q_1 q_2 q_3 + \\
&\quad + q_5 + q_3 + q_1.
\end{aligned}$$

Dla $n = 6$ otrzymamy

$$\begin{aligned} & f(q_1, q_2, q_3, q_4, q_5, q_6) = \\ & \quad = q_1q_2q_3q_4q_5q_6 + \\ & \qquad + q_3q_4q_5q_6 + q_1q_4q_5q_6 + q_1q_2q_5q_6 + q_1q_2q_3q_6 + q_1q_2q_3q_4 + \\ & \qquad + q_5q_6 + q_3q_6 + q_3q_4 + q_1q_6 + q_1q_4 + q_1q_2 + \\ & \qquad + 1. \end{aligned}$$

Zauważmy, że jeśli z iloczynu $q_1q_2 \cdot \ldots \cdot q_n$ usuniemy wszystkie możliwe pary, to otrzymamy składnik równy 1.

Wszystkie otrzymane w ten sposób wyrażenia są oczywiście symetryczne:

$$f(q_1, q_2, \ldots, q_n) = f(q_n, \ldots, q_2, q_1).$$

DOWODY

(1) Ciąg

$$|t_{n+1}|, \quad |t_n|, \quad \ldots, \quad |t_1|, \quad |t_0|$$

powstaje z tych samych iloczynów wziętych w odwrotnej kolejności. Zatem

$$|t_{n+1}| = f(q_n, \ldots, q_2, q_1) = f(q_1, q_2, \ldots, q_n) = r_0 = p.$$

Następnie

$$t_n d = t_n r_1 \equiv r_n = 1 \pmod{p}.$$

Mnożąc tę kongruencję przez d otrzymamy

$$t_n d^2 \equiv d \pmod{p},$$

czyli

$$t_n \equiv -d \pmod{p}.$$

Ponieważ $|t_n| < p$, więc

$$t_n = d \;\text{ lub }\; t_n = p - d.$$

Gdyby $t_n = p - d$, to algorytm Euklidesa dla pary liczb $|t_{n+1}| = p$ i $|t_n| = p - d$ dałby ciąg

$$p, \quad p - d, \quad , d, \quad \ldots$$

Ten ciąg byłby jednak dłuższy od ciągu reszt r_i, co jest niemożliwe. Zatem $t_n = -d$, czyli $|t_n| = d$. Ciąg

$$|t_{n+1}|, \quad |t_n|, \quad \ldots, \quad |t_1|, \quad |t_0|$$

zaczyna się od p i d, więc jest identyczny z ciągiem

$$r_0, \quad r_1, \quad \ldots, \quad r_{n+1}.$$

(2) Wiemy, że

$$t_{2i} < 0.$$

Ponieważ $t_n = -d < 0$, więc liczba n jest parzysta.

(3) Ilorazy otrzymane w algorytmie Euklidesa dla pary liczb p i d są takie same jak ilorazy otrzymane dla pary liczb $|t_{n+1}|$ i $|t_n|$. Zatem ciąg ilorazów jest symetryczny.

(4) Wiemy, że

$$t_i r_1 \equiv r_i \pmod{p},$$

czyli

$$t_i d \equiv r_i \pmod{p}.$$

Stąd

$$t_i^2 d^2 \equiv r_i^2 \pmod{p},$$

czyli

$$t_i^2 + r_i^2 \equiv 0 \pmod{p}.$$

(5) Ze wzoru Eulera wynika, że

$$r_{\frac{n}{2}} = f(q_{\frac{n}{2}+1}, \ldots, q_n)$$

oraz

$$r_{\frac{n}{2}} = |t_{\frac{n}{2}+1}| = f(q_1, \ldots, q_{\frac{n}{2}}).$$

Łatwo można zauważyć, że każdy składnik po wymnożeniu

$$f(q_{\frac{n}{2}+1}, \ldots, q_n) \cdot f(q_1, \ldots, q_{\frac{n}{2}})$$

występuje w wyrażeniu

$$f(q_1, \ldots, q_n).$$

Zatem

$$r_{\frac{n}{2}}^2 \leq p.$$

Ponieważ liczba p jest pierwsza, więc

$$r_{\frac{n}{2}}^2 < p.$$

Wystarczy więc dowieść, że

$$r_{\frac{n}{2}-1}^2 \geq p.$$

Ale w tym celu zauważmy, że każdy składnik występujący w wyrażeniu

$$f(q_1, \ldots, q_n)$$

występuje również w wyrażeniu powstałym z wymnożenia

$$f(q_{\frac{n}{2}}, \ldots, q_n) \cdot f(q_1, \ldots, q_{\frac{n}{2}+1}).$$

Stąd

$$p \leq r^2_{\frac{n}{2}-1}.$$

To ostatecznie kończy dowód.

ONG-SCHNORR-SHAMIR SIGNATURE SCHEME

System ten został opublikowany przez Onga, Schnorra i Shamira w roku 1984. A oto opis systemu.

(1) Osoba zaufana wybiera liczbę n postaci $n = pq$; zakładamy, że liczbę n trudno rozłożyć na czynniki.

(2) Każdy użytkownik, który chce podpisać wiadomość m, wybiera losowo tajną liczbę s i oblicza

$$k = s^2 \mathbin{\mathbf{mod}} n.$$

Liczbę (klucz publiczny) k przekazuje osobie zaufanej, która publikuje otrzymane klucze publiczne.

(3) Aby podpisać wiadomość m, wspomniany użytkownik publikuje rozwiązanie (x, y) kongruencji

$$x^2 - ky^2 \equiv m \pmod{n}. \qquad (*)$$

(4) Każdy może łatwo sprawdzić poprawność podpisu złożonego pod wiadomością m.

(5) Użytkownik znający tajny klucz s może łatwo znaleźć rozwiązanie kongruencji $(*)$: wybiera losowo liczbę r, a następnie oblicza

$$x = 2^{-1}(mr^{-1} + r) \mathbin{\mathbf{mod}} n, \quad y = (2s)^{-1}(mr^{-1} - r) \mathbin{\mathbf{mod}} n.$$

Bezpieczeństwo systemu zależy od tego, czy osoba postronna potrafi rozwiązać rozwiązać kongruencję $(*)$, znając wyłącznie k, m i n i nie znając s.

ZŁAMANIE SYSTEMU

Niedługo po opublikowaniu system ten został złamany przez Pollarda. Oto szkic rozumowania Pollarda.

Kongruencję

$$x^2 - ky^2 \equiv m \pmod{n}$$

redukujemy do kongruencji

$$x^2 - k_1 y^2 \equiv m \pmod{n}$$

dla k_1 takiego, że $|k_1| \leq \sqrt{4|k|/3}$. Po niewielkiej liczbie takich redukcji dochodzimy do kongruencji postaci

$$x^2 \pm y^2 \equiv m \pmod{n}.$$

Kongruencję

$$x^2 - y^2 \equiv m \pmod{n}$$

rozwiązujemy trywialnie, znajdując rozwiązanie układu kongruencji

$$x - y \equiv m \pmod{n}, \quad x + y \equiv 1 \pmod{n}.$$

Kongruencję

$$x^2 + y^2 \equiv m \pmod{n}$$

rozwiązujemy znajdując liczbę pierwszą p taką, że

$$p \equiv m \pmod{n} \text{ oraz } p \equiv 1 \pmod{4}.$$

Następnie znajdujemy rozkład liczby p na sumę kwadratów:

$$x^2 + y^2 = p \equiv m \pmod{n}.$$