

## Лекция 7.

### 2.8. Построение больших простых чисел.

Алгоритмы построения больших простых чисел носят рекурсивный характер. Чтобы построить большое простое число, нужно построить возрастающую последовательность простых чисел. Число нужного размера появляется в конце этой последовательности, члены же её принадлежат последовательности уменьшающихся промежутков, с построения которых начинается алгоритм. Эта конструкция вместе со всеми пояснениями обсуждается в настоящем параграфе.

Сначала мы рассмотрим следующую задачу: Для каждого заданного натурального  $N \geq 11$  **построить простое число  $Q$ , лежащее на промежутке  $2^{N-1} \leq Q < 2^N$ .**

1. Для каждого числа  $N \leq 18$  требуемое простое число можно найти непосредственно. Например, на промежутке от  $2^{17}$  до  $2^{18}$  их имеется 10749 штук, и каждое составное число этого промежутка делится на какое нибудь простое  $p < 2^9 = 512$ . Для нахождения простого числа в нужном маленьком промежутке (правый конец не превосходит  $2^{18}$ ) можно воспользоваться алгоритмом пробных делений, см. подраздел 2.10.1.

Далее будет найдена последовательность целых чисел  $a_k, b_k, q_k$  таких что

$$q_k \text{ - простые, } 2^{a_k} \leq q_k < 2^{b_k}, \quad a_k = b_k - 1, \quad 1 \leq k \leq m, \quad b_m = N.$$

Тогда  $Q = q_m$  - искомое простое число.

2. Пусть  $N$  - натуральное число, превосходящее 18. Построим последовательность целых чисел, начинающуюся с  $N$ , в которой за числом  $\ell$  следует число  $\lceil \frac{\ell}{2} \rceil + 1$ . Легко видеть, что эта последовательность убывает при  $\ell \geq 4$ . Более того, в ней обязательно встретится целое число из промежутка  $11 \leq \ell \leq 18$ . Кроме того, не трудно проверить, что при любом  $\ell \geq 19$  выполняется неравенство  $\lceil \frac{\ell}{2} \rceil + 1 \geq 11$ .

Например, для  $N = 1024$  эта последовательность имеет вид

$$N = 1024, \quad 513, \quad 258, \quad 130, \quad 66, \quad 34, \quad 18.$$

Итак,

*с какого бы числа  $N$  ни начиналась указанная последовательность, она обязательно содержит единственное число из промежутка  $11 \leq$*

$x \leq 18$ . Начиная с этого числа, перенумеруем в обратном порядке все члены последовательности буквами  $b_1, b_2, \dots, b_m$ . Тогда  $11 \leq b_1 \leq 18$  и  $b_m = N$ . Для каждого индекса  $k$ ,  $1 \leq k \leq m$ , положим  $a_k = b_k - 1$ .

3. Простое число  $q_1$  выбирается так, как это указано в пункте 1, ведь  $11 \leq b_1 \leq 18$  и  $2^{a_1} < q_1 < 2^{b_1}$ . Например, в случае  $b_1 = 11$  на роль  $q_1$  может быть взято любое из 137 простых чисел, лежащих на отрезке от 1024 до 2048, скажем 1933.

Предположим теперь, что  $k \geq 2$  и простое число  $q_{k-1}$  уже построено. Для краткости обозначим

$$\ell = b_k, \quad q = q_{k-1}. \quad \text{Тогда} \quad a_k = \ell - 1, \quad b_{k-1} = \left\lceil \frac{\ell}{2} \right\rceil + 1, \quad a_{k-1} = \left\lceil \frac{\ell}{2} \right\rceil.$$

Число  $q_k$  будет строиться с помощью Следствия (2) из теоремы (7), см. также алгоритм (5). При этом  $F = q$ ,  $R = 2t$ , где  $t$  - некоторое натуральное, оно будет выбираться с помощью испытаний, и

$$c = 2tq + 1 = FR + 1. \quad (2.25)$$

Для того, чтобы число  $c$  попало в нужный интервал, т.е.  $2^{\ell-1} < c < 2^\ell$ , на параметр  $t$  накладываются условия

$$\frac{2^{\ell-1}}{2q} \leq t \leq \frac{2^\ell - 1}{2q}. \quad (2.26)$$

Действительно, в этих ограничениях имеем

$$c = 2tq + 1 \geq 2^{\ell-1} + 1 > 2^{\ell-1} = 2^{a_k}$$

и

$$c = 2tq + 1 < 2^\ell = 2^{b_k}.$$

Строгое неравенство выполняется в силу того, что  $c$  нечётно, а  $2^\ell$  чётное число. Не трудно проверить, что для любых положительных чисел  $u < v$  на интервале  $u < t < v$  лежит не менее  $v - u - 1$  целых чисел  $t$ . Согласно неравенству (2.26) можно утверждать, что количество целых  $t$ , для которых число  $c$  принадлежит интервалу  $2^{\ell-1} < c < 2^\ell$  не меньше, чем  $\frac{2^\ell - 1}{2q} - 1$ . Например, при  $b_1 = 11$ ,  $\ell = b_2 = 19$  и  $q = 1933$  на промежутке  $2^{18} < c < 2^{19}$  имеется 10 простых чисел вида  $2tq + 1$ . Самые маленькие из них соответствуют числам  $t = 65, 72, 75, 77, 81$ . При  $t = 77$  имеем  $c = 297683$ .

4. Как правило, строящиеся большие простые числа должны иметь случайный характер. Поэтому на интервале от  $2^{\ell-1}$  до  $2^\ell$  выбирается

случайное число  $x$ , а затем от него начинается отсчёт чисел  $t$ , соответствующие им числа  $s$  проверяются на простоту. Как только попадётся простое  $s$ , оно выбирается в качестве следующего простого  $q_k$ . Если при очередном  $t$  соответствующее число  $s$  выйдет за верхний предел заданного промежутка, следующее значение  $t$  принимается равным наименьшему возможному значению и вычисления продолжаются.

Выберем псевдослучайное целое число  $x$  на промежутке  $2^{\ell-1} \leq x < 2^\ell$  и положим  $t = \lceil \frac{x}{2q} \rceil$ . Определим также число  $s$  равенством (2.25).

Из неравенства  $x \geq 2^{\ell-1}$  следует (2.25).

5. Если  $t$  оказалось настолько большим, что

$$2tq + 1 \geq 2^\ell,$$

то полагаем  $t = \lceil \frac{2^{\ell-1}}{2q} \rceil$ . Если с помощью следствия 2 при  $F = q$  и  $R = 2t$  выбирая псевдослучайные числа  $b$  на промежутке от 2 до  $s - 2$ , удаётся доказать, что число  $s$  простое, то выбираем  $q_k = s$ . Для того, чтобы с помощью следствия 2 можно было доказать простоту числа  $s$ , параметры  $R$  и  $F$  должны быть связаны неравенством  $R \leq 4F + 2$ , которое может быть записано в виде  $t \leq 2q + 1$ . Используя границы для  $t$  и  $q$ , находим

$$q^2 > 2^{2\lceil \frac{\ell}{2} \rceil} \geq 2^\ell > 2qt \quad \text{и} \quad t < \frac{1}{2}q < 2q + 1,$$

так что следствие 2 применимо.

Если  $k = t$ , алгоритм завершает свою работу. В противном случае увеличиваем  $k$  на единицу и переходим в пункт 3 к следующему промежутку  $[a_k, b_k)$ .

6. Если же  $s$  оказывается составным, то параметр  $t$  увеличивается на 1 и алгоритм переходит в пункт 5.

Следующий далее алгоритм, по заданным числам  $L$  и  $N$  строит пары простых чисел  $P, Q$  с условиями

$$Q|(P - 1), \quad 2^{L-1} \leq P < 2^L, \quad 2^{N-1} \leq Q < 2^N.$$

Мы ограничимся здесь только формальным описанием его основных моментов.

**Алгоритм 7.** Даны натуральные числа  $L$  и  $N, L > N$ .

**Построить** "случайные" простые числа  $P, Q$  битовой длины  $L$  и  $N$  с условием  $Q|(P - 1)$ .

1. С помощью описанного выше алгоритма построить "случайное" простое число  $Q$  с длиной записи  $N$  битов.

2. Построить последовательность простых чисел  $p_k$ , связанных соотношениями

$$p_k = 2tQp_{k-1} + 1, \quad 1, \dots, m, \quad 2^{L-1} \leq p_m < 2^L.$$

Для проверки на простоту чисел  $s = 2tQp_{k-1} + 1$  использовать следствие 2 с параметрами  $F = p_{k-1}$ ,  $R = 2tQ$ .

3. Положить  $P = p_m$ .

## 2.9. Первообразные корни и дискретное логарифмирование.

Пусть  $p$  – простое нечётное число. Множество классов вычетов целых чисел по модулю  $p$  составляет поле  $\mathbb{F}_p$ , а ненулевые элементы этого поля образуют циклическую мультипликативную группу порядка  $p - 1$ . Согласно малой теореме Ферма выполняется сравнение  $a^{p-1} \equiv 1 \pmod{p}$ . Наименьшее натуральное число  $d$  с условием  $a^d \equiv 1 \pmod{p}$  называется *показателем* числа  $a$  по модулю  $p$ . Например, показатель числа 1 по модулю  $p$  равен 1, а показатель  $-1$  по нечётному модулю  $p$  равен 2. Целое число  $a$ , не делящееся на  $p$ , называется *первообразным корнем по модулю  $p$* , если его показатель равен  $p - 1$ . Например, показатель числа 5 по модулю 23 равен 22. Мы проверим это позже. Число 5 есть первообразный корень по модулю 23. Его класс вычетов по модулю 23 порождает мультипликативную группу всех ненулевых классов вычетов по модулю 23.

В 1801 году К.Ф. Гаусс опубликовал два доказательства следующей теоремы.

**Теорема 8.** Для каждого простого нечетного числа  $p$  существуют первообразные корни по модулю  $p$ . Количество не сравнимых друг с другом по модулю  $p$  первообразных корней равно  $\varphi(p - 1)$ , где  $\varphi(n)$  – функция Эйлера.

Для практического нахождения первообразных корней удобно пользоваться следующим утверждением.

**Теорема 9.** Целое число  $g$ , не делящееся на простое нечетное  $p$ , будет первообразным корнем по модулю  $p$  в том и только том случае, если для любого простого числа  $q$ , делящего  $p - 1$ , выполняется

$$g^{\frac{p-1}{q}} \not\equiv 1 \pmod{p}.$$

*Доказательство.* Обозначим буквой  $d$  показатель числа  $g$  по модулю  $p$ . Разделив  $p - 1$  на  $d$  с остатком, получим

$$p - 1 = d \cdot u + v, \quad 0 \leq v < d.$$

Неравенство  $v > 0$  в силу малой теоремы Ферма и сравнений

$$1 \equiv g^{p-1} = (g^d)^u \cdot g^v \equiv g^v \pmod{p}$$

противоречит определению  $d$ . Значит  $v = 0$  и  $d \mid (p - 1)$ . Если  $d < p - 1$ , то найдется такое простое число  $q$ , что  $d \mid \frac{p-1}{q}$ . Но тогда  $g^{\frac{p-1}{q}} = (g^d)^{\frac{p-1}{qd}} \equiv 1 \pmod{p}$ , вопреки условию. Значит,  $d = p - 1$ , и это завершает доказательство теоремы.  $\square$

Если известны все простые делители числа  $p - 1$ , то проверка условий теоремы 9 выполняется достаточно быстро с помощью алгоритма, изложенного в параграфе 2.3.

Справедливы сравнения по модулю 23

$$5^{11} = 5 \cdot 25^5 \equiv 5 \cdot 2^5 \equiv -1 \pmod{23}$$

и  $5^2 \equiv 2 \pmod{23}$ . Применяя теорему 9 к простому числу  $p = 23$  и  $s = 5$ , заключаем, что 5 есть первообразный корень по модулю 23.

Множество первообразных корней при заданном  $p$  достаточно велико, поэтому выбирая числа  $g$  случайным образом на промежутке  $0 < g < p$ , можно с большой вероятностью попасть на первообразный корень и доказать это с помощью теоремы 9.

Если  $g$  – первообразный корень по модулю  $p$ , то числа

$$1, g, g^2, \dots, g^{p-2} \tag{2.27}$$

имеют разные наименьшие неотрицательные остатки. Действительно, если бы нашлись целые числа  $0 \leq u < v \leq p - 2$ , для которых  $g^u$  и  $g^v$  имеют одинаковые остатки, то  $g^u \equiv g^v \pmod{p}$ . В силу взаимной простоты чисел  $g$  и  $p$  мы получили бы

$$1 \equiv g^{v-u} \pmod{p}.$$

Но это невозможно, так как  $0 < v - u < p - 1$ , а показатель  $g$  равен  $p - 1$ .

Множество (2.27) состоит из  $p - 1$  чисел. Поэтому для каждого целого числа  $a$ , не делящегося на  $p$ , найдется единственное целое  $k$ , удовлетворяющее условиям

$$a \equiv g^k \pmod{p}, \quad 0 \leq k < p - 1. \tag{2.28}$$

Число  $k$  называют *индексом*  $a$  по модулю  $p$  при основании  $g$ . Это определение и формулируемые ниже свойства индексов напоминают свойства обычных логарифмов действительных чисел. Поэтому иногда в современной литературе индексы называют дискретными логарифмами, а процесс их нахождения – дискретным логарифмированием. Используются также обозначения  $\text{ind}_g a$ ,  $\text{Log}_g a$ . Указание на первообразный корень иногда опускается. Итак, имеем

$$a \equiv g^{\text{ind}_g a} \pmod{p}, \quad 0 \leq \text{ind}_g a < p - 1. \quad (2.29)$$

Свойства индексов описывают следующие простые утверждения.

1. Если целые числа  $a, b$  не делятся на  $p$ , то

$$\text{ind}_g ab \equiv \text{ind}_g a + \text{ind}_g b \pmod{p - 1}.$$

2. Если  $a, b$  – первообразные корни по модулю  $p$ , то для любого числа  $c$ , не делящегося на  $p$ , выполняется сравнение

$$\text{ind}_b c \equiv \text{ind}_b a \cdot \text{ind}_a c \pmod{p - 1}.$$

Сформулируем теперь задачу дискретного логарифмирования по простому нечетному модулю.

Дано простое число  $p$ . Для заданных чисел  $a, b \in \mathbb{Z}$ , не делящихся на  $p$ , требуется решить сравнение

$$b^x \equiv a \pmod{p}. \quad (2.30)$$

Решение этой задачи очень трудоемко в вычислительном отношении. Не случайно в конце практически всех учебников по элементарной теории чисел приводятся таблицы индексов (дискретных логарифмов).

Лучшие из известных алгоритмов дискретного логарифмирования по простому модулю  $p$ , использующие вычисления в полях алгебраических чисел, требуют  $O(e^{2(\ln p)^{1/3}(\ln \ln p)^{2/3}})$  арифметических операций. Впрочем, эта оценка условна, ибо опирается на ряд недоказанных, но весьма правдоподобных гипотез теории чисел.

Рекордный по величине простого числа  $p$  результат в задаче дискретного логарифмирования был установлен в 2016 году. Группе европейских математиков и программистов удалось сосчитать дискретный логарифм по модулю простого числа, записываемого 232 десятичными цифрами (768 битами). Применённый ими алгоритм требует достаточно глубоких познаний в теории алгебраических чисел, и мы его здесь не объясняем.

Излагаемый здесь метод решения сравнения (2.30) требует  $O(p^{1/2} \ln p)$  арифметических операций. Он был придуман в 1962г. А.О.Гельфондом и

в русскоязычной литературе называется "метод согласования". Аналогичный метод был предложен в 1971г. Д.Шенксом. Его английское название может быть переведено как "метод больших и малых шагов".

**Лемма 1.** Пусть  $p$  — простое нечетное число и пусть  $H = [\sqrt{p}] + 1$ . Тогда для каждого целого числа  $d$ ,  $1 \leq d < p$ , найдутся целые числа  $u, v$ , удовлетворяющие условиям

$$1 \leq u, v \leq H, \quad Hu - v = d.$$

*Доказательство.* Положим

$$u = \left[ \frac{d}{H} \right] + 1, \quad v = Hu - d.$$

Тогда

$$\frac{d}{H} < u \leq \frac{d}{H} + 1,$$

откуда видим, что, во-первых,

$$0 < u < \frac{p}{\sqrt{p}} + 1 = \sqrt{p} + 1,$$

а во-вторых,  $0 < Hu - d \leq H$ . Остается воспользоваться тем, что числа  $u$  и  $v$  целые.  $\square$

**Алгоритм 8** (Алгоритм Гельфонда). Данные: Простое число  $p \geq 3$ , первообразный корень  $g$  по модулю  $p$ , число  $b \in \mathbb{Z}$ ,  $p \nmid b$ .

Найти: Решение сравнения  $g^x \equiv b$ .

1. Вычислить  $H = [\sqrt{p}] + 1$ .
2. Положить  $c \equiv g^H \pmod{p}$ ,  $1 \leq c < p$ .
3. Составить два набора чисел

$$S_1 = \{c^u \pmod{p} : 1 \leq u \leq H\}, \quad S_2 = \{bg^v \pmod{p} : 1 \leq v \leq H\}.$$

4. Упорядочить по возрастанию оба набора  $S_1$  и  $S_2$ . Найти совпавшие элементы этих наборов, то есть такие числа  $u, v$ , для которых

$$c^u \equiv bg^v \pmod{p}. \quad (2.31)$$

5. Положить

$$\text{ind}_g b = Hu - v. \quad (2.32)$$

Решение заданного сравнения имеет вид  $x \equiv \text{ind}_g b \pmod{p-1}$ .

Пересечение множеств  $S_1$  и  $S_2$  не пусто, так как сравнение (2.31) равносильно представлению (2.32), а последнее, согласно лемме 1, существует всегда.

Вычисления в пунктах 1 и 2 требуют  $O(\ln p)$  арифметических операций. Вычисления в пункте 3 требуют  $O(H \ln p) = O(\sqrt{p} \ln p)$  арифметических операций. Для упорядочения каждого из множеств  $S_1, S_2$  нужно  $O(H \ln H) = O(\sqrt{p} \ln p)$  арифметических операций. Для нахождения одинаковых элементов в упорядоченных множествах  $S_1, S_2$  нужно  $O(H) = O(\sqrt{p})$  арифметических операций. Общее же количество операций в алгоритме Гельфонда равно  $O(\sqrt{p} \ln p)$ .

**Пример.** Решить сравнение

$$2^x \equiv 23 \pmod{37}. \quad (2.33)$$

**Решение.** Так как  $36 = 2^2 \cdot 3^2$  и

$$2^{18} \equiv -1 \not\equiv 1 \pmod{37}, \quad 2^{12} \equiv 26 \not\equiv 1 \pmod{37},$$

то 2 - первообразный корень по модулю 37 и, значит данное сравнение разрешимо. В соответствии с алгоритмом вычисляем  $H = [\sqrt{37}] + 1 = 7$  и  $c = 2^7 \equiv 17 \pmod{37}$ . Множества  $S_1$  и  $S_2$  состоят из чисел  $c^n \pmod{37}$ , и  $b \cdot a^n \pmod{37}$  при  $n = 1, 2, \dots, 7$ , т.е.

$$S_1 = \{17, 30, 29, 12, 19, 27, 15\}, \quad S_2 = \{9, 18, 36, 35, 33, 29, 21\}.$$

Эти два множества имеют общий элемент 29, который стоит на  $u = 3$  месте в первом множестве и на  $v = 6$  месте во втором множестве. Таким образом,  $\text{ind}_2 23 = 7 \cdot 3 - 6 = 15$ , а решение сравнения (2.33) имеет вид  $x \equiv 15 \pmod{37}$ .

Если в сравнении (2.30) число  $b$  не является первообразным корнем по модулю  $p$ , то это сравнение можно переписать в равносильном виде

$$x \cdot \text{ind}_g b \equiv \text{ind}_g a \pmod{p-1}. \quad (2.34)$$

Индексы чисел  $a, b$  можно вычислить с помощью алгоритма Гельфонда. После этого можно решить линейное сравнение (2.34), см. параграф ???. Все найденные числа  $x$  составят решения (2.30). Если сравнение (2.34) не имеет решений, то не будет их иметь и сравнение (2.30).

Алгоритм Гельфонда, конечно, быстрее полного перебора, но он работает недопустимо медленно, чтобы решать задачу дискретного логарифмирования для простых чисел размером в 250 десятичных цифр.

**Конец седьмой лекции.**



## Лекция 8.

### 2.10. Задача разложения целых чисел на множители.

В этой главе будут рассматриваться простейшие методы разложения целых чисел на простые сомножители, т.е. методы поиска для заданного целого  $N > 1$  простых чисел  $p_1, \dots, p_r$  таких, что

$$N = p_1^{k_1} \cdots p_r^{k_r}, \quad k_j \geq 1, \quad k_j \in \mathbb{Z}.$$

При этом будет предполагаться, что разлагаемое число  $N$  составное, в чем можно убедиться с помощью тестов из параграфа 2.5.

Достаточно уметь решать более простую задачу о разложении целого числа на два меньших множителя, т.е. задачу о решении в целых числах  $a > 1, b > 1$  уравнения  $N = ab$ . Действительно, в этом случае выполняются неравенства  $a < N, b < N$ , можно разложить на два меньших множителя каждое из чисел  $a, b$ , и продолжать далее эту процедуру, пока такое разложение будет возможным, т.е. до тех пор, пока все сомножители не станут простыми.

Существующие алгоритмы разложения чисел на множители могут быть распределены на группы в зависимости от количества арифметических операций, которые алгоритм требует для своей работы.

1) *Экспоненциальные* алгоритмы используют  $O(N^c)$  арифметических операций. Здесь  $c$  — положительная постоянная.

2) *Субэкспоненциальные* алгоритмы требуют  $O(e^{c(\ln N)^\alpha (\ln \ln N)^\beta})$  арифметических операций. Здесь  $\alpha, \beta, c$  — положительные постоянные,  $\alpha + \beta = 1$ . Заметим, что при  $\alpha = 1$ , т.е.  $\beta = 0$ , оценка совпадает с оценкой сложности экспоненциальных алгоритмов.

Для наиболее быстрого из субэкспоненциальных алгоритмов, так называемого *метода решета числового поля*, или метода просеивания в числовых полях имеем  $\alpha = \frac{1}{3}, \beta = \frac{2}{3}$ .

Алгоритмы полиномиальной сложности,  $\alpha = 0, \beta = 1$ , для задачи факторизации не известны, и весьма вероятно, что их не существует.

Деятельность по разложению чисел на множители сочетает в себе черты инженерной науки, поскольку во многом опирается на допущения, основанные на опыте и не имеющие теоретических обоснований, а с другой стороны она сродни искусству, так как зачастую продолжительность работы алгоритма и результат зависят от удачного выбора параметров.

### 2.10.1. Алгоритм пробных делений.

Пусть  $d_1 < d_2 < \dots$  — последовательность целых чисел, содержащая все простые числа. Алгоритм пробных делений состоит в последовательном делении  $N$  на числа  $d_1, d_2, \dots$ , не превосходящие  $\sqrt{N}$ . Если  $N$  составное число, то оно имеет простой делитель  $p \leq \sqrt{N}$  и потому будет разложено на множители.

Этот алгоритм часто используется для нахождения всех простых делителей числа  $N$ , не превосходящих некоторой заданной границы  $B$ .

Последовательность  $d_i$  может совпадать с множеством простых чисел. Но в этом случае необходим алгоритм, строящий все простые числа до заданной границы. Иногда бывает проще использовать последовательности, содержащие и составные числа, но менее сложные в реализации.

**Пример 1.** Каждое простое число  $p > 6$  удовлетворяет одному из двух сравнений  $p \equiv 1 \pmod{6}$  или  $p \equiv 5 \pmod{6}$ . Поэтому последовательность

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 25, \dots$$

содержащая все числа вида  $6n \pm 1, n \in \mathbb{N}$ , включает в себя и все простые числа. Но не только их. Например, она содержит 25 и 91, и многие другие составные числа. Но простое правило порождения этой последовательности облегчает поиск делителей  $N$ :

$$d_1 = 2, d_2 = 3, d_3 = 5, \quad d_{2k} = d_{2k-1} + 2, \quad d_{2k+1} = d_{2k} + 4, \quad k \geq 2.$$

При таком выборе  $d_i$  алгоритм вообще говоря требует  $O(N^{1/2})$  арифметических операций и  $O(1)$  памяти. Конечно, он очень медленный, но позволяет быстро отсеивать составные числа, имеющие малые простые делители.

**Пример 2.** В качестве модуля в сравнениях вместо 6 можно взять число  $m = 2 \cdot 3 \cdot 5 = 30$ . Все простые числа  $p > 5$  будут содержаться в восьми прогрессиях с разностью 30, начинающихся с чисел 1, 7, 11, 13, 17, 19, 23, 29, т.е.

$$30k + 7, 30k + 11, 30k + 13, 30k + 17, 30k + 19, 30k + 23, 30k + 29, 30k + 31.$$

Соответствующая последовательность  $d_i$  порождается формулами

$$\begin{aligned} d_1 &= 2, d_2 = 3, d_3 = 5, & d_{8k+4} &= d_{8k+3} + 2, \\ d_{8k+5} &= d_{8k+4} + 4, & d_{8k+6} &= d_{8k+5} + 2, \\ d_{8k+7} &= d_{8k+6} + 4, & d_{8k+8} &= d_{8k+7} + 2, \\ d_{8k+9} &= d_{8k+8} + 4, & d_{8k+10} &= d_{8k+9} + 6, \\ d_{8k+11} &= d_{8k+10} + 2, & d_{8k+12} &= d_{8k+11} + 6, \quad k \geq 0. \end{aligned}$$

Соответствующая последовательность  $\{d_i\}$  будет содержать меньшую долю составных чисел. Если  $m = \prod_{p \leq r} p$ , то количество арифметических прогрессий, составляющих эту последовательность равно  $\varphi(m)$ , а доля чисел из  $\{d_i\}$  в натуральном ряде есть  $\frac{\varphi(m)}{m} = \prod_{p \leq r} \left(1 - \frac{1}{p}\right)$ . При  $m = 6$  она равна  $\frac{1}{3}$ , а при  $m = 30$  имеем  $\frac{1}{4}$ .

### 2.10.2. $\rho$ — метод Полларда.

Пусть  $f(x)$  — "достаточно случайный" многочлен. Выберем "случайно"  $x_1 \in \mathbb{Z}, 1 < x_1 < N$ , и рассмотрим последовательность

$$x_{k+1} \equiv f(x_k) \pmod{N}, \quad 0 \leq x_{k+1} < N, \quad k \geq 1, \quad (2.35)$$

Так как количество классов вычетов по модулю  $N$  не превосходит  $N$ , то существуют такие индексы  $i, j, 0 \leq i, j < N$ , что  $x_i \equiv x_j \pmod{N}$  и значит, последовательность  $x_k \pmod{N}$  зацикливается. Период этой последовательности не всегда начинается с самого начала, она может иметь предпериодическую часть. Символически такую последовательность можно изобразить в виде греческой буквы  $\rho$ . Подходящая снизу ножка этой буквы соответствует предпериодической части последовательности, она переходит в замкнутую петлю, соответствующую периодической части. Эта аналогия и дала название алгоритму.

Поллард обнаружил, что для простых  $p$ , как правило, длина периода и предпериодическая часть последовательности

$$x_{k+1} \equiv f(x_k) \pmod{p}, \quad k \geq 1,$$

ограничены сверху величиной  $c\sqrt{p}$ , где  $c$  — некоторая константа.

Идея алгоритма состоит в том, чтобы последовательно вычислять наибольшие общие делители  $(x_{2i} - x_i, N)$ ,  $i = 1, 2, 3, \dots$ . Если  $p$  — простой делитель  $N$ ,  $\ell$  — длина периода и  $a$  — длина предпериодической части последовательности  $x_{k+1} \equiv f(x_k) \pmod{p}$ , то для номера  $i$ , удовлетворяющего условиям

$$i > a, \quad \ell | i, \quad (2.36)$$

числа  $x_i$  и  $x_{i+\ell}$  сравнимы друг с другом по модулю  $p$ . Мы не знаем чисел  $\ell, i$ , но знаем, что разность  $2i - i = i$  делится на  $\ell$ , поэтому  $x_{2i} \equiv x_i \pmod{p}$ , так что  $(x_{2i} - x_i, N) > 1$ . Ясно, что существует число  $i$ , удовлетворяющее условиям (2.36) и неравенству  $i \leq a + \ell = O(\sqrt{p})$ . Конечно, может случиться, что  $N | x_{2i} - x_i$ . Тогда нужно выбирать иное начальное значение  $x_1$ .

Если  $p$  — наименьший простой делитель  $N$ , то  $p \leq N^{1/2}$ , так что  $i = O(N^{1/4})$ .

В следующем ниже алгоритме вычисляются пары  $\{x_i, x_{2i}\}, i \geq 1$ . Для нахождения следующей пары  $\{x_{i+1}, x_{2i+2}\}$  нужно вычислить

$$\begin{aligned} x_{i+1} &\equiv f(x_i) \pmod{N}, & x_{2i+1} &\equiv f(x_{2i}) \pmod{N}, \\ x_{2i+2} &\equiv f(x_{2i+1}) \pmod{N}, \end{aligned}$$

т.е. выполнить  $O(1)$  арифметических операций.

**Алгоритм 9.** Дано: составное число  $N$ . Найти: нетривиальный делитель  $N$ .

1. Выбрать случайно  $x_1 \in \mathbb{Z}, 1 < x_1 < N$ , и положить

$$x = f(x_1) \pmod{N}, \quad y = f(x) \pmod{N}.$$

2. Вычислить  $d = (y - x, N)$ . Если  $1 < d < N$  алгоритм останавливается, нетривиальный делитель  $d$  числа  $N$  найден.

3. Если  $d = N$ , перейти в п. 1.

4. Положить

$$x = f(x) \pmod{N}, \quad z = f(y) \pmod{N}, \quad y = f(z) \pmod{N}$$

и перейти в пункт 2 алгоритма.

Если все удачно сложится, то нетривиальный делитель числа  $N$  будет найден за  $O(p^{1/2})$  арифметических операций, где  $p$  — наименьший простой делитель  $N$ , т.е. за  $O(N^{1/4})$  арифметических операций.

В силу оценки сложности  $O(p^{1/2})$  алгоритм удобен для нахождения не очень больших делителей  $p$  числа  $N$ , если они есть.

В качестве  $f(x)$  обычно выбираются многочлены вида  $x^2 + a$ . Например, можно взять  $f(x) = x^2 + 1$  или  $f(x) = x^2 - 1$ . В то же время выбор  $f(x) = x^2 - 2$  и  $x_1 = 2$  не очень удачен, так как в этом случае последовательность  $x_k$  имеет период 1.

Приведем теперь некоторые правдоподобные соображения в пользу того, что длина периода и предпериодическая часть последовательности  $x_{k+1} \equiv f(x_k) \pmod{p}$  оцениваются сверху величиной  $O(p^{1/2})$ .

### 2.10.3. Парадокс дней рождения.

Пусть  $\mathcal{R}$  — множество, состоящее из  $r$  элементов. Из принципа ящиков Дирихле следует, что в любой последовательности  $z_1, z_2, \dots, z_{r+1}, z_i \in \mathcal{R}$ , обязательно найдутся два одинаковых элемента. Но можно проверить,

что выбирая случайным образом множество  $z_1, z_2, \dots, z_m$  с теми же условиями на  $z_i$  при достаточно большом, но всё же существенно меньшем  $r$  значении  $m$ , можно с вероятностью, большей  $\frac{1}{2}$  попасть на набор, имеющий два одинаковых элемента. Этот кажущийся парадокс носит название "парадокс дней рождения" по причине, о которой мы расскажем немного позже, а сейчас докажем, при некоторых естественных предположениях, указанную оценку на вероятность "хорошей" выборки.

Будем выбирать случайным образом наборы

$$\bar{z} = \{z_1, \dots, z_m\}, \quad z_i \in \mathcal{R}, \quad (2.37)$$

предполагая, что все они равновероятны. Какова вероятность того, что выбранная таким образом совокупность  $z_1, z_2, \dots, z_m$ , содержит по крайней мере два одинаковых элемента?

Каждое значение  $z_i$  может равняться одному из  $r$  элементов множества  $\mathcal{R}$ . Значит количество последовательностей вида (2.40) равно  $r^m$ . Сколько же среди них последовательностей с различными элементами. Попробуем построить такие последовательности. На первом месте может стоять любое из возможных  $r$  значений для  $z_1$ . На втором месте может стоять лишь  $r - 1$  значений, отличных от стоящего на первом месте. На третьем месте может стоять лишь  $r - 2$  значения, отличных от первых двух. Действуя так далее мы сможем построить  $r(r - 1)(r - 2) \dots (r - m + 1)$  последовательностей, состоящих из  $m$  различных значений. Легко видеть, что каждая такая последовательность будет учтена в этом процессе. Следовательно, доля последовательностей с различными значениями  $z_i$  будет равна

$$\mu = \frac{r(r - 1) \dots (r - m + 1)}{r^m}.$$

Для оценки  $\mu$  сверху перепишем

$$\mu = \prod_{k=1}^{m-1} \left(1 - \frac{k}{r}\right) \quad \text{и} \quad \ln \mu = \sum_{k=1}^{m-1} \ln\left(1 - \frac{k}{r}\right). \quad (2.38)$$

При любом действительном  $x$ ,  $0 \leq x < 1$  справедливо неравенство

$$\ln(1 - x) \leq -x. \quad (2.39)$$

Для его доказательства рассмотрим непрерывную на промежутке  $[0; 1)$  функцию  $g(x) = x + \ln(1 - x)$ . Так как производная  $g'(x) = \frac{-x}{1-x}$  отрицательна на интервале  $(0; 1)$ , то  $g(x)$  убывает на множестве  $0 \leq x < 1$ , и, значит, на этом множестве выполняется неравенство (2.39).

Взяв  $x = \frac{k}{r}$ , находим  $\ln(1 - \frac{k}{r}) < -\frac{k}{r}$ ,  $1 \leq k < m$  и из (2.38)

$$\ln \mu < -\sum_{k=1}^{m-1} \frac{k}{r} = -\frac{m^2 - m}{2r} < -\frac{(m-1)^2}{2r}.$$

Выберем теперь  $m = \lceil \sqrt{2r \ln 2} \rceil + 1$ . Тогда  $(m-1)^2 = \lceil \sqrt{2r \ln 2} \rceil^2 \geq 2r \ln 2$  и  $\ln \mu < -\ln 2$ . Так доказано неравенство  $\mu < \frac{1}{2}$ . Заметим, что при  $r \geq 5$  выполняется  $m < r$ .

Если увеличить константу  $\sqrt{2 \ln 2}$  в определении  $m$ , оценка  $\mu$  уменьшится и, значит, вероятность попасть на набор, имеющий по крайней мере два равных элемента, станет больше  $\frac{1}{2}$ . Например, в случае  $m = \lceil 4\sqrt{r} \rceil$  вероятность попасть на выборку, имеющую 2 или более одинаковых элементов, превосходит 0,9996....

Вернёмся теперь к последовательности (2.35) и возьмём постоянную в определении  $m$  большую, чем  $\sqrt{2 \ln 2}$ . Из сказанного выше следует, что для "хорошего" многочлена  $f(x)$  и быть может после нескольких случайных выборов числа  $x_1$  мы за  $O(\sqrt{p} \ln p)$  арифметических операций сможем найти последовательность  $x_1, x_2, \dots, x_m$ , имеющую два равных числа. Длины предпериода  $a$  и периода  $\ell$  этой последовательности удовлетворяют неравенству  $a + \ell \leq M$ . Действительно,  $a$  есть номер первого из двух равных чисел, и  $a + \ell$  - номер второго из них чисел. Это объясняет наблюдение Полларда о длине предпериода и периода последовательности в  $\rho$ -методе Полларда.

Конечно, это рассуждение не является доказательством, но оно даёт правдоподобное объяснение, почему  $\rho$  - метод Полларда достаточно быстро на практике находит небольшие простые делители составных чисел.

Заметим также, что скорость работы этого алгоритма может быть увеличена, если наибольший общий делитель  $(x_{2i} - x_i, N)$  вычислять не на каждом шаге. Например, можно для последовательности  $i = 0, d, 2d, \dots$  вычислять  $(\prod_{k=i+1}^{i+d} (x_{2k} - x_k), N)$ , выбрав  $d$  не очень большим.

Теперь мы обсудим алгоритм, также основанный на парадоксе дней рождения и позволяющий найти коллизию у любой хеш-функции  $H(x)$ . Он носит вероятностный характер. Обозначим буквой  $r$  количество всех различных значений функции  $H(\bar{x})$ , положим также  $m = \lceil \sqrt{2r} \rceil + 1$ . Не трудно проверить, что при  $r \geq 6$  выполняется неравенство  $m < r$ . Выберем случайно различные аргументы  $x_1, \dots, x_m$  функции  $H(\bar{x})$  и вычислим хеш-значения

$$H(x_1), \quad \dots, \quad H(x_m). \quad (2.40)$$

Если среди этих значений есть одинаковые, то коллизия построена. При выбранном  $m$  справедливы неравенства  $(m-1)^2 > \lceil \sqrt{2r} \rceil^2 > 2r$  и  $\ln \mu < -1$ , так что в противном случае доля последовательностей без одинаковых элементов не превосходит  $e^{-1}$ . Поэтому  $\mu < e^{-1} = 0,367\dots$ , и количество последовательностей с различными хеш-значениями существенно меньше половины всех последовательностей. Это доказывает, что если все значения хеш-функции равновероятны, то при случайном выборе набора аргументов  $x_1, x_2, \dots, x_M$  вероятность получить последовательность значений, имеющую равные элементы, превосходит  $1 - e^{-1} > \frac{1}{2}$ . Если все элементы выбранной последовательности хеш-значений оказались различными, можно выбрать вторую последовательность. Если опять не повезло, можно выбрать третью последовательность. Вероятность выбрать последовательность с равными элементами после  $k$  шагов не меньше  $1 - e^{-k}$ . С ростом  $k$  она очень быстро приближается к 1. К сожалению, этот гарантированно работающий алгоритм требует слишком много времени. Например, в случае  $r = 2^{256}$  имеем  $m = \lceil \sqrt{2r} \rceil + 1 > 2^{128}$ . Последовательность из  $2^{128}$  хеш-значений имеет огромную длину.

Рассмотрим ещё одну ситуацию, которая собственно и дала название корневому понижению оценки числа арифметических операций при нахождении в последовательности равных элементов. Представим себе, что студенческая группа или просто группа знакомых собралась на вечеринку по какому-то поводу. В группе 23 человека и требуется определить, какова вероятность, что среди приглашенных есть двое, дни рождения которых приходятся на одну дату. На первый взгляд кажется, что эта вероятность очень маленькая. Но рассмотрим ситуацию чуть подробнее. Если год, в котором происходит это событие не високосный, т.е. состоит из  $p = 365$  дней, то вероятность выбрать 23 различные даты из 365 равна

$$\begin{aligned} \mu &= \prod_{k=1}^{22} \left(1 - \frac{k}{365}\right) = 0,492703\dots = \\ &= \frac{36997978566217959340182499134166757044383351847256064}{75091883268515350125426207425223147563269805908203125}. \end{aligned}$$

Последнее значение точное. Таким образом, любой случайный набор из 23 дат невисокосного года с вероятностью большей  $\frac{1}{2}$  содержит две одинаковых даты.

**Конец восьмой лекции.**

## Лекция 9.

### 2.11. Эллиптические кривые.

Многие математические открытия, как и открытия в физике, химии, биологии, других науках, основаны на эксперименте. Один из величайших математиков Л. Эйлер, живший в XVIII веке, нашел и доказал множество фундаментальных математических фактов. Он обнаружил их, выполняя вручную колоссальные по объёму вычисления. Сейчас для этого часто используется компьютер.

Уравнение

$$y^2 = x^3 + ax + b, \quad (2.41)$$

с коэффициентами  $a, b$ , удовлетворяющими условию  $4a^3 + 27b^2 \neq 0$ , определяет на плоскости кривую, называемую *эллиптической*. Неравенство означает, что многочлен  $x^3 + ax + b$  не имеет кратных корней. Докажем этот факт.

**Лемма 2.** *Многочлен  $f(x) = x^3 + ax + b$  имеет кратные корни в том и только том случае, когда  $4a^3 + 27b^2 = 0$ .*

*Доказательство.* Обозначим корни производной  $f'(x) = 3x^2 + a$  буквами  $x_1$  и  $x_2 = -x_1$ . Для того, чтобы эти числа были действительными необходимо и достаточно выполнение неравенства  $a \leq 0$ . Подставим эти корни в многочлен  $f(x)$  и перемножим получившиеся числа. В результате, учитывая, что  $x_1^2 = -\frac{a}{3}$ , получим

$$\begin{aligned} f(x_1)f(-x_1) &= (x_1^3 + ax_1 + b)(-x_1^3 - ax_1 + b) = b^2 - (x_1^3 + ax_1)^2 = \\ &= b^2 - x_1^2(x_1^2 + a)^2 = b^2 + \frac{4a^3}{27}. \end{aligned}$$

Многочлен  $f(x)$  имеет кратный корень в том и только том случае, когда он и его производная имеют общий корень, т.е.  $f(x_1) = 0$  или  $f(-x_1) = 0$ , что по доказанному равносильно равенству  $b^2 + \frac{4a^3}{27} = 0$  или равенству  $4a^3 + 27b^2 = 0$ .  $\square$

Если многочлен  $f(x)$  имеет кратные корни, то кривая может быть параметризована рациональными функциями и её исследование сильно упрощается.

Например, для многочлена  $x^3 - 3x + 2 = (x - 1)^2(x + 2)$  выполняется тождество

$$(t^3 - 3t)^2 = (t^2 - 2)^3 - 3(t^2 - 2) + 2,$$



означающее, что при любом  $t$  формулы  $x = t^2 - 2, y = t^3 - 3t$  задают некоторую точку на кривой  $y^2 = x^3 - 3x + 2$ . Легко увидеть, что так может быть получена любая точка  $(x, y)$  этой кривой. Соответствующее значение переменной  $t$  определяется формулой  $t = \frac{y}{x-1}$ . Теперь достаточно проверить равенства

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

$$\left(\frac{y}{x-1}\right)^3 - 3\frac{y}{x-1} = \frac{y}{(x-1)^3}(y^2 - (x-1)^2(x+2) + (x-1)^3) = y.$$

Точка кривой  $(1, 0)$ , для которой выражение  $\frac{y}{x-1}$  не определено, получается по указанным формулам при  $t = \sqrt{3}$ .

Кривая предполагается размещённой на действительной плоскости. В дальнейшем будут рассматриваться такие кривые и над конечными полями. Именно этот случай используется в криптографии.

Эллиптическими называются и другие кривые, уравнения которых с помощью взаимно-обратных преобразований, задаваемых рациональными функциями от переменных, могут быть преобразованы к виду (2.41).

Название эллиптические кривые происходит от названия эллиптические функции. Эти функции не выражаются через элементарные, но хорошо изучены, благодаря важной роли, которую они играют в приложениях. Через них, в частности, можно выразить длину дуги эллипса, с чем и связано их название. Эллиптические кривые параметризуются эллиптическими функциями так же как окружность параметризуется синусом и косинусом угла. В дальнейшем будет предполагаться, что коэффициенты  $a, b$  уравнения (2.41) суть рациональные числа. Это условие нужно потому, что дальше будут рассматриваться решения этого уравнения в целых или рациональных числах.

Теория, описывающая свойства решений в рациональных числах или, как мы будем писать в дальнейшем *рациональных решений*, имеет большую историю и богата глубокими результатами. Тем не менее в ней есть естественные открытые вопросы, т.е. вопросы, ответы на которые не известны. Вот некоторые из них.

**1. Есть ли решение?** До сих пор не известен алгоритм, позволяющий для любого данного уравнения вида (2.41) с рациональными коэффициентами  $a, b$  определить, есть у него решение в рациональных числах или нет.

**2. Как найти решение, если известно, что оно существует?** То, что ответ на этот вопрос достаточно сложен, показывает следующий

пример. Известно, что уравнение  $y^2 = x^3 + 877x$  имеет бесконечно много рациональных решений. Решение этого уравнения с наименьшими знаменателями имеет вид

$$x = \frac{375494528127162193105504069942092792346201}{6215987776871505425463220780697238044100},$$

$$y = \frac{256256267988926809388776834045513089648669153204356603464786949}{490078023219787588959802933995928925096061616470779979261000}.$$

**3. Конечно или бесконечно множество решений?**

**4. Какова структура множества решений?**

### 2.11.1. Метод касательных

Ещё древние греки знали, что по одному рациональному решению уравнения (2.41) иногда можно построить ещё одно решение, затем ещё и т.д. Метод, которым они пользовались сейчас называется метод касательных. Древнегреческий учёный Диофант Александрийский, живший в III веке, сочинил книгу "Арифметика", посвящённую решению в целых или рациональных числах алгебраических уравнений и систем уравнений. По его имени алгебраические уравнения, которые требуется разрешить в целых или рациональных числах, называются диофантовыми.

В книге кн. VI, под номером 18 рассматривается следующая задача:

*Найти такой прямоугольный треугольник, чтобы его площадь, увеличенная на гипотенузу, образовала куб, а периметр был квадратом.*

Здесь под кубом имеется в виду куб некоторого рационального числа, а под квадратом - квадрат другого рационального числа.

Решая эту задачу, Диофант выбирает один из катетов треугольника равным 2. Так делать нельзя, если мы хотим найти все решения задачи, но, если нужно найти какое-нибудь решение, такой выбор допустим, и он существенно упрощает задачу. Если обозначить сторону квадрата буквой  $y$ , а длину ребра куба буквой  $x$ , то данные в условии соотношения приводят к уравнению

$$y^2 = x^3 + 2, \tag{2.42}$$

причём  $x, y$  рациональны. Уравнение (2.42) имеет решения  $x = -1, y = \pm 1$ , но по смыслу задачи  $x$  и  $y$  должны быть положительными рациональными числами. Поэтому очевидные решения не подходят. Диофант предлагает следующий замечательный приём. Добавим к уравнению (2.42) ещё одно уравнение  $2y = 3x + 5$ , т.е. рассмотрим систему

уравнений

$$\begin{cases} y^2 = x^3 + 2 \\ 2y = 3x + 5. \end{cases} \quad (2.43)$$

Легко видеть, что второе уравнение системы задаёт прямую - касательную к кривой (2.42), см. левый рис.1, проведённую в известной нам рациональной точке  $(-1, 1)$ .

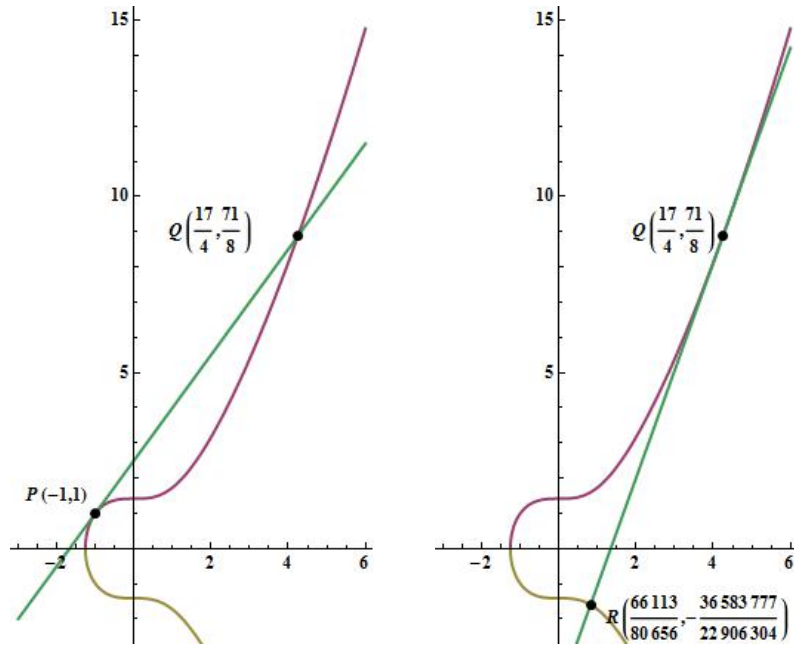


Рис. 2.1. Метод касательных

Решения системы (2.43) соответствуют точкам пересечения эллиптической кривой (2.42) с касательной. Решая систему (2.43), находим

$$(3x + 5)^2 = 4(x^3 + 2)$$

Два корня этого уравнения нам известны до его решения. Это  $x_1 = x_2 = -1$ , ведь по построению прямая, добавленная к уравнению эллиптической кривой (2.42), касается этой кривой в точке  $(-1, 1)$ . Пользуясь, например, теоремой Виета, находим  $x_1 + x_2 + x_3 = \frac{9}{4}$ , откуда следует  $x_3 = \frac{17}{4}$  и  $y_3 = \frac{3x_3 + 5}{2} = \frac{71}{8}$ . Второй катет  $z$  искомого треугольника удовлетворяет равенству

$$2 + z + \sqrt{z^2 + 4} = \left(\frac{71}{8}\right)^2.$$

Отсюда получаем  $z^2 + 4 = \left(\frac{4913}{64} - z\right)^2$  и находим второй катет искомого прямоугольного треугольника  $z = \frac{24121185}{628864}$ .

Заметим, что Диофант не пользовался геометрическими соображениями и работал только с уравнениями.

Процесс построения рациональных точек можно продолжить и далее. Проведём касательную через найденную точку  $(\frac{17}{4}, \frac{71}{8})$ . Эта прямая пересечёт кривую (2.42) ещё в одной рациональной точке  $(\frac{66113}{80656}, -\frac{36583777}{22906304})$ . Так можно построить бесконечную последовательность рациональных точек на кривой (2.42).

### 2.11.2. Метод секущих.

Следующий шаг в этой истории был сделан спустя много сотен лет двумя великими учёными И. Ньютоном (1643-1727), и Л. Эйлером (1707-1783).

Ньютон не опубликовал своё наблюдение, оно было найдено в его документах после смерти. Эйлер же в 1768 году опубликовал на русском языке учебник "Универсальная арифметика", сыгравший важную роль в преподавании алгебры и арифметики.

Обратимся к левому рисунку

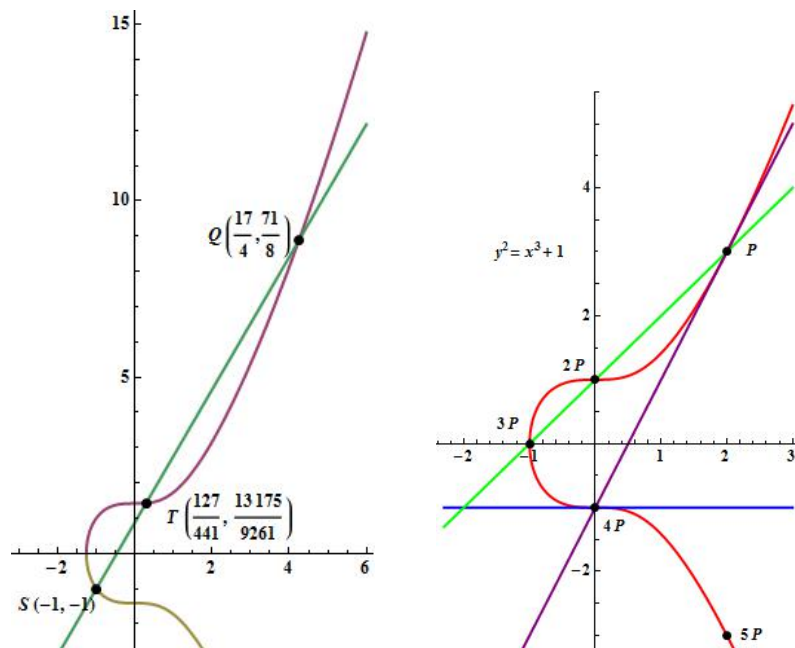


Рис. 2.2. Метод секущих

На нём нарисована кривая (4.9) и через рациональные точки  $(-1, -1)$  и  $(\frac{17}{4}, \frac{71}{8})$  проведена прямая линия

$$y + 1 = \frac{79}{42}(x + 1) \quad (2.44)$$

или  $y = \frac{79x + 37}{42}$ . Решая уравнение  $(\frac{79x + 37}{42})^2 = x^3 + 2$ , находим корни  $x_1 = -1, x_2 = \frac{17}{4}$  - абсциссы точек, через которые мы проводили

прямую, и  $x_3 = \frac{127}{441}$  - абсциссу третьей точки пересечения кривой (2.42) и построенной прямой. Подставляя  $\frac{127}{441}$  вместо  $x$  в правую часть равенства (2.44) находим ординату третьей точки пересечения  $y_3 = \frac{13175}{9261}$ . Проводя секущие и касательные через уже построенные точки, мы можем получать всё новые и новые рациональные точки на кривой (2.42).

Конечно, метод секущих и касательных может быть применён к любой кривой вида (2.41). Нужно лишь знать хотя бы одну рациональную точку на ней. Но как уже было написано, именно этот вопрос не решен до сих пор. Не известен алгоритм, который говорил бы для любой кривой вида (2.41), есть на ней рациональная точка или нет. Нет и алгоритма, который находил бы рациональную точку, если известно, что она есть.

Рассмотрим ещё один любопытный пример. На правом рисунке 2 красным цветом изображена кривая, задаваемая уравнением

$$y^2 = x^3 + 1. \quad (2.45)$$

Легко проверить, что это уравнение имеет следующие 5 решений в рациональных числах

$$(-1, 0), \quad (0, -1), \quad (0, 1), \quad (2, -3), \quad (2, 3).$$

На правом рисунке эти пять точек обозначены  $P, 2P, 3P, 4P, 5P$ . Причина таких обозначений выяснится чуть позже. Здесь же давайте обратим внимание на то, что касательная к любой из этих точек пересекает кривую в одной из точек этого же множества. Например, касательная в точке  $4P$  имеет уравнение  $y = -1$ . Подставляя  $y = -1$  в уравнение (2.45), приходим к уравнению  $1 = x^3 + 1$ , имеющему единственный корень  $x = 0$  кратности 3. Новая рациональная точка таким способом не получается. Вертикальная прямая, проходящая через точки  $P$  и  $5P$ , имеет уравнение  $x = 2$ . Она пересекает кривую (2.45) только в двух точках  $(2, \pm 3)$ . Третья точка пересечения отсутствует. Формально считают, что любая вертикальная прямая пересекает кривую (2.41) на бесконечности. Так что в рассматриваемом случае третья точка пересечения кривой (2.45) и прямой  $y = 2$  находится в бесконечности, т.е. бесконечно удалена от начала координат. Можно ввести так называемые проективные координаты, в них все точки кривой, в том числе и бесконечно удалённая, станут равноправными. Бесконечно удалённую точку часто обозначают буквой  $O$ , похожей на ноль.

Итак, касательные и секущие не позволяют построить новую рациональную точку на кривой (2.45). Причина в том, что никаких других рациональных точек кроме пяти указанных  $(-1, 0), (0, \pm 1), (2, \pm 3)$  на

этой кривой не существует. Этот непростой факт доказал в 1738 году Л.Эйлер.

### 2.11.3. Сложение точек на эллиптической кривой

Посмотрим на описанный выше метод проведения секущих и касательных не как на метод размножения рациональных точек, а как на некоторую операцию, выполняемую на множестве точек эллиптической кривой.

Пусть  $P$  и  $Q$  две точки кривой (2.41). Проведём через  $P$  и  $Q$  прямую и третью точку её пересечения с кривой отразим симметрично относительно оси  $OX$ . Получившуюся точку назовём суммой  $P$  и  $Q$ . Будем обозначать эту точку  $P + Q$ , см. рисунок 3.

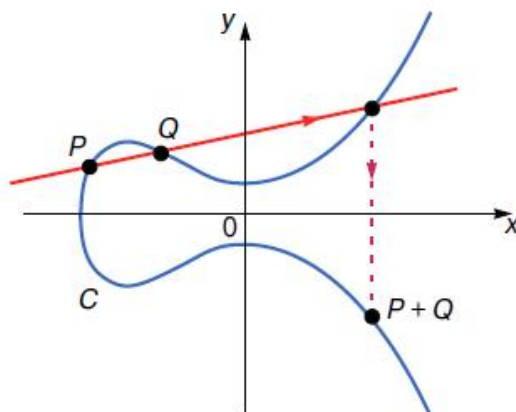


Рис. 2.3. Операция сложения точек на эллиптической кривой.

Если точки  $P$  и  $Q$  одинаковы, проведём касательную в точке  $P$  и точку пересечения с кривой отразим симметрично относительно оси  $OX$ . Получившуюся точку будем обозначать  $2P$ , см. правый рис. 4. Легко проверить с помощью таких же соображений, что  $P + O = P$ , а также  $P + R = O$ , см. левый рисунок 4.

Чтобы поупражняться в сложении точек, можно проверить на правом рисунке 2, что

$$P+P = 2P, \quad P+2P = 3P, \quad P+4P = 5P, \quad P+5P = O, \quad 4P+4P = 2P.$$

Операция сложения в координатах задается следующим образом.

**Теорема 10.** Если  $(x_1, y_1)$  - координаты точки  $P$  и  $(x_2, y_2)$  - координаты точки  $Q$ , причём обе точки отличны от  $O$ , а также  $Q \neq -P$ , то

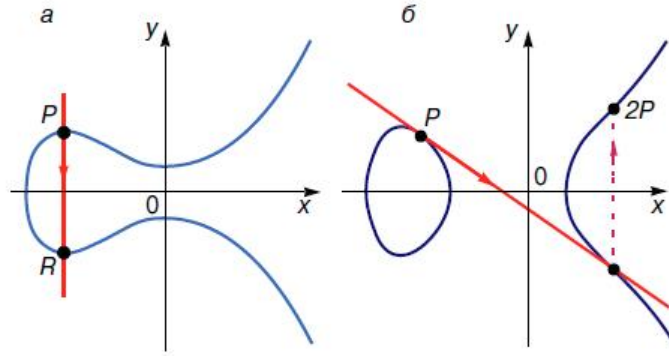


Рис. 2.4. Операция сложения точек на эллиптической кривой.

координаты  $(x_3, y_3)$  точки  $P + Q$  можно вычислить по формулам

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2, \\ y_3 = \lambda(x_1 - x_3) - y_1, \end{cases} \quad \text{где } \lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{если } x_2 \neq x_1, \\ \frac{3x_1^2 + a}{2y_1}, & \text{если } x_2 = x_1 \text{ и } y_2 = y_1. \end{cases} \quad (2.46)$$

Кроме того, по определению для любой точки  $P$  выполняются равенства  $P + O = O + P = P$  и  $P + (-P) = O$ , где  $-P = (x_1, -y_1)$ .

*Доказательство.* В случае  $x_2 \neq x_1$  тангенс угла наклона прямой, проходящей через точки  $P, Q$  равен  $\frac{y_2 - y_1}{x_2 - x_1} = \lambda$  и уравнение этой прямой имеет вид

$$y - y_1 = \lambda(x - x_1). \quad (2.47)$$

В этом легко убедиться, подставив в уравнение (2.47) координаты точек  $P$  и  $Q$ . Из определения точки  $P + Q$  следует, что числа  $(x_3, -y_3)$  удовлетворяют уравнениям (2.41) и (2.47). Выразив переменную  $y$  через  $x$  из уравнения (2.47) и подставив получившееся выражение в уравнение (2.41), найдём кубическое уравнение

$$(y_1 + \lambda(x - x_1))^2 = x^3 + ax + b,$$

корнями которого являются абсциссы трёх точек пересечения кривой (2.41) и прямой (2.47), т.е. числа  $x_1, x_2, x_3$ . Полученное уравнение может быть переписано также в виде  $x^3 - \lambda^2 x^2 + \dots = 0$ , где опущены слагаемые имеющие степень по переменной  $x$  меньшую чем 2. Согласно теореме Виета сумма  $x_1 + x_2 + x_3$  должна быть равна  $\lambda^2$ , откуда следует выражение для  $x_3$  из равенств (2.46). Точка  $(x_3, -y_3)$  лежит на прямой  $PQ$ . Из (2.47) теперь следует равенство  $-y_3 - y_1 = \lambda(x_3 - x_1)$  и далее выражение для  $y_3$  из (2.47). В первом случае теорема доказана.

Рассмотрим теперь второй случай, когда  $Q = P$ . В этом случае вместо секущей прямой  $PQ$  рассматривается касательная к кривой (2.41),

проведённая в точке  $P$ . В случае  $y_1 = 0$  имеем равенство  $P = -P$  и находим  $P + P = P + (-P) = O$ . Теорема доказана и в этом случае.

Далее считаем  $Q = P$  и  $y_1 \neq 0$ . В этом случае можно найти на плоскости маленькую окрестность точки  $P = (x_1, y_1)$ , в которой уравнение (2.41) определяет  $y = y(x)$  как однозначную функцию от переменной  $x$ , а именно  $y(x) = \sqrt{x^3 + ax + b}$ , где ветвь квадратного корня выбрана так, чтобы значение  $\sqrt{x_1^3 + ax_1 + b}$  равнялось  $y_1$ . Эта функция удовлетворяет в малой окрестности точки  $x_1$  тождеству  $y(x)^2 = ax^3 + ax + b$ . Дифференцируя это тождество по переменной  $x$ , находим  $2y(x)y'(x) = 3x^2 + a$ . Подставляя в полученное тождество  $x = x_1$  и пользуясь равенством  $y(x_1) = y_1$ , находим

$$y'(x_1) = \frac{3x_1^2 + a}{2y_1} = \lambda.$$

Так что в этом случае число  $\lambda$  равно тангенсу угла наклона касательной к кривой, определённой равенством (2.41), проведённой в точке  $P$ . Далее все рассуждения проводятся в точности так же, как и в первом случае, с учётом равенства  $Q = P$ .  $\square$

Введённое выше определение суммы точек на первый взгляд кажется странным, однако оно обладает рядом удобных и полезных свойств: точка  $O$  играет такую же роль, как и ноль в множестве целых чисел, для каждой точки  $P$  есть единственная обратная, обозначаемая  $-P$ , т.е. точка с условием  $P + (-P) = O$ , операция сложения коммутативна, т.е.  $P + Q = Q + P$  для любых двух точек  $P, Q$  и ассоциативна, т.е. для любых трёх точек  $P, Q, R$  эллиптической кривой выполняется равенство  $(P + Q) + R = P + (Q + R)$  (доказательство последнего факта не просто). Множества с такой операцией называют *группами*.

В 1901г. выдающийся французский учёный А. Пуанкаре (1854-1912) высказал предположение, что *все рациональные точки эллиптической кривой могут быть получены из конечного числа проведением хорд и касательных*. По другому: *группа рациональных точек имеет конечное число образующих*.

Выше мы показали, что для кривой  $y^2 = x^3 + 1$ , см. правый рис. 2, все рациональные точки могут быть получены с помощью сложения из одной точки  $P = (2, 3)$ . Здесь, конечно, используется результат Эйлера, что других рациональных точек, кроме пяти указанных и точки  $O$  на этой кривой не лежит.

Гипотезу Пуанкаре доказал в 1922 году английский математик Л. Морделл (1888-1972).



Приведём ещё один пример.

*Кривая  $y^2 = x^3 + 877x$  имеет бесконечно много рациональных точек.  
Точка*

$$x = \frac{375494528127162193105504069942092792346201}{6215987776871505425463220780697238044100},$$

$$y = \frac{256256267988926809388776834045513089648669153204356603464786949}{490078023219787588959802933995928925096061616470779979261000}.$$

*есть её единственная образующая.*

Известен пример эллиптической кривой с числом образующих, не меньшим 28, но точное число образующих в этом случае не известно. Кривые с числом образующих большим 28 в настоящее время не найдены. Тем не менее предполагается, что

*Существуют эллиптические кривые со сколь угодно большим числом независимых образующих.*

Но эта гипотеза до сих пор не доказана. Далее мы не будем встречаться с эллиптическими кривыми, определёнными над полем рациональных или действительных чисел.

#### 2.11.4. Эллиптические кривые над конечными полями.

Пусть  $p > 3$  - простое число. Рассмотрим теперь уравнение (2.41), предполагая, что его коэффициенты принадлежат  $\mathbb{F}_p$ . Множество пар чисел  $x, y$  из  $\mathbb{F}_p$ , удовлетворяющих уравнению (2.41), также будем называть эллиптической кривой, добавляя при этом "над полем  $\mathbb{F}_p$ ". Эта "кривая" состоит из конечного множества точек, ведь каждая переменная  $x$  и  $y$  может принимать не более  $p$  значений.

Например, кривая, определённая уравнением  $y^2 = x^3 + 2$  над множеством  $\mathbb{F}_{11}$  состоит из точек

$$(1, 5), (7, 2), (6, 3), (9, 7), (10, 1), (4, 0), (10, 10), (9, 4), (6, 8), (7, 9), (1, 6) \quad (2.48)$$

и ещё точки  $O$ , которую по аналогии тоже можно называть бесконечно удалённой точкой. Указанная эллиптическая кривая, определённая над полем  $\mathbb{F}_{11}$ , состоит из 12 точек.

То же уравнение над полем  $\mathbb{F}_7$  определяет кривую

$$(0, 3), (0, 4), (3, 1), (3, 6), (5, 1), (5, 6), (6, 1), (6, 6), \quad (2.49)$$

состоящую из 9 точек: 8 указанных выше точек и ещё бесконечно удалённая точка  $O$ .

Точки кривой, определённой над конечным полем, тоже можно складывать. В этом случае, конечно, нельзя проводить касательные, но можно воспользоваться формулами (2.46), выполняя все вычисления в поле  $\mathbb{F}$  по указанным в (2.46) формулам. И в этом случае точка  $O$  будет играть роль нуля, а операция сложения будет коммутативной и ассоциативной.

Например, если в случае кривой  $y^2 = x^3 + 2$  обозначить  $P = (1, 5)$ , то все точки в (2.48) будут иметь вид

$$P, 2P, 3P, \dots 11P,$$

а  $12P = O$ .

А для каждой точки  $Q$  кривой (2.49) выполняется равенство  $3Q = O$ .

**Конец девятой лекции и части 1 .**

## Глава 3.

# Криптографические примитивы.

### Лекция 10.

Методы шифрования можно разделить на два типа: на *симметричные* и *асимметричные*.

При симметричном шифровании и для зашифровки, и для расшифровки сообщения применяется один и тот же ключ. Поэтому для безопасности передачи зашифрованного сообщения по открытым каналам необходимо предварительно обеим сторонам в тайне выбрать единый ключ, и лишь после этого начинать обмен информацией. При таком способе шифрования возникают очевидные трудности, связанные с необходимостью либо предварительной встречи, либо поиска надежного курьера, которому можно доверить секретный ключ. В частности, организовать надежный конфиденциальный канал связи для большого числа абонентов при таком способе шифрования довольно затруднительно.

Общая идея асимметричных методов шифрования заключается в том, что для зашифровки сообщения применяется один ключ, а для расшифровки — другой. Ключ для зашифровки открыто передается по любому каналу связи, в то время как ключ для расшифровки держится в секрете. Эти ключи, как правило, являются элементами некоторых алгебраических структур и связаны друг с другом некоторым соотношением. Но параметры этих структур и соотношение подбираются такими, чтобы время, требуемое для построения секретного ключа по открытому, было несравнимо больше того времени, которое нужно для построения открытого ключа по секретному.

Асимметричные методы шифрования обычно медленнее симметричных. Поэтому, если требуется передать большой объем информации, очень часто тело сообщения шифруют при помощи какого-нибудь симметричного алгоритма, а использующийся при этом ключ — при помощи асимметричного. Подобного рода схемы называются гибридными.

### 3.1. Алгоритм Диффи–Хеллмана обмена ключами.

Первой работой, в которой была реализована идея асимметричного криптографического преобразования, является работа [10]. Авторы этой работы использовали тот факт, что задача дискретного логарифмирования в мультипликативной группе  $F_p^*$  поля  $F_p$ , где  $p$  — простое число, существенно сложнее задачи возведения в степень.

Отметим, что это обстоятельство можно очень легко использовать для защиты произвольного ресурса, будь то компьютер, или канал связи, от несанкционированного доступа. Для этого нужно выбрать достаточно большое простое число  $p$  и какую-нибудь образующую  $g$  группы  $F_p^*$ . Далее, каждый пользователь “придумывает” себе секретный пароль  $t \in \mathbb{Z}$  и вычисляет элемент  $g^t$  поля  $F_p$ , который система и будет воспринимать как идентификатор данного пользователя. И каждый раз, когда пользователь решает войти в систему, он вводит свой секретный пароль  $t$ , система вычисляет  $g^t$  и сравнивает результат с идентификатором пользователя. Возможность взлома такой системы зависит от способности злоумышленника вычислять дискретный логарифм по основанию  $g$  от идентификатора пользователя, то есть выбор достаточно больших  $p$  может обеспечить необходимую безопасность. Отметим однако, что образующую  $g$  нужно выбирать не слишком маленькой, чтобы исключить возможность найти  $g$  перебором.

Алгоритм же, о котором сейчас пойдет речь, позволяет двум лицам, общающимся по незащищенному каналу связи, без предварительного обмена секретной информацией выработать общий ключ, который будет известен только им двоим. Предположим, абонент  $A$  и абонент  $B$  выбрали простое число  $p$  и образующую  $g$  группы  $F_p^*$ . Случайным образом абонент  $A$  выбирает секретный ключ  $a \in \mathbb{Z}$ ,  $2 \leq a \leq p - 2$ , а абонент  $B$  — секретный ключ  $b \in \mathbb{Z}$ ,  $2 \leq b \leq p - 2$ .

**Алгоритм 1.** *Данные:* Простое число  $p$ , первообразный корень  $g$  по модулю  $p$ , секретный ключ  $a$  абонента  $A$  и секретный ключ  $b$  абонента  $B$ .

*Найти:* Общий ключ  $k \in F_p^*$ , известный только абонентам  $A$  и  $B$ .

1. Абоненту  $A$  вычислить  $x = g^a$  и передать результат абоненту  $B$ .
2. Абоненту  $B$  вычислить  $y = g^b$  и передать результат абоненту  $A$ .
3. Абоненту  $A$  положить  $k = y^a$ ,
4. Абоненту  $B$  положить  $k = x^b$ .

Корректность данного алгоритма следует из равенства  $k_B = (g^a)^b = (g^b)^a = k_A$ . Открытыми ключами здесь являются элементы  $g^a$  и  $g^b$ . Они

передаются по незащищенному каналу связи, поэтому могут стать известными третьим лицам. Но если число  $p$  было выбрано достаточно большим, то для того, чтобы по известным  $g^a$  и  $g^b$  при помощи существующих на данный момент алгоритмов найти секретные числа  $a$  и  $b$  (и, соответственно,  $g^{ab}$ ), потребуется очень много времени, что и обеспечивает требуемую безопасность. Аналогично, абоненты  $A$  и  $B$  не смогут за приемлемое время вычислить секретные ключи друг друга, то есть каждый из них при смене собеседника может не менять свой секретный ключ.

В описанном виде схема Диффи-Хеллмана уязвима относительно так называемой "атаки посередине", в которой абонент - нарушитель  $C$  выступает в качестве посредника между  $A$  и  $B$ .

$$A \xleftarrow{z=g^c} \xrightarrow{x=g^a} C \xleftarrow{y=g^b} \xrightarrow{z=g^c} B$$

Если злоумышленнику удалось убедить абонента  $A$  воспринять его как абонента  $B$ , а абонента  $B$  — соответственно, как  $A$ , он, наладив с каждым из них связь, при помощи алгоритма Диффи-Хеллмана, создаёт секретный ключ  $g^{ac}$  для обмена информацией с  $A$  и секретный ключ  $g^{bc}$  для обмена информацией с  $B$ . В результате он получает полный контроль за перепиской  $A$  и  $B$ , включая возможность искажения и передачи ложной информации. Чтобы предотвратить подобную возможность, используют так называемые алгоритмы аутентификации.

Отметим также, что алгоритм 1 по своей структуре несколько отличается от изложенных ранее. Обычно предполагается, что у алгоритма ровно один исполнитель, и по этой причине команды имеют безадресный характер. Что же касается алгоритмов типа алгоритма 1, то они предполагают как минимум двоих исполнителей. Поэтому такие алгоритмы носят название *протоколов*.

## 3.2. Хеш-функции.

### 3.2.1. Конструкция Дамгорда-Меркля.

Напомним, что *отображение*

$$h : \{0, 1\}^* \mapsto \{0, 1\}^n, n \in \mathbb{N}$$

называется хеш-функцией, если оно *однонаправленно*, не имеет коллизий и найти для него второй прообраз очень сложно. Рассмотрим отображение

$$F : \{0, 1\}^n \times \{0, 1\}^k \mapsto \{0, 1\}^n, k \in \mathbb{N}$$

и для конечного сообщения  $M$  построим хеш-значение  $h(M)$ :

- 1) Дополнить сообщение  $M$  до длины, кратной  $k$  бит, и разбить полученный набор бит на блоки длины  $k$ :  $M \rightarrow M^{(0)} || \dots || M^{(N-1)}$ ,  
 $M^{(i)} \in \{0, 1\}^k, 0 \leq i < N, i \in \mathbb{N}$ .
- 2) Пусть  $H^{(0)}$  – заранее фиксированная константа
- 3) Для всех  $0 < i \leq N$  вычислить  $H^{(i)} := F(H^{(i-1)}, M^{(i-1)})$ .
- 4) Положить значение  $h(M) := H^{(N)}$ .

Функция  $F$  называется *сжимающей функцией*.

Далее мы будем подробно рассматривать функцию хеширования SHA-256. Алгоритм вычисления её значений может использоваться для хеширования любого сообщения  $M$ , имеющего длину меньшую, чем  $2^{64}$  бита. В основе его лежит конструкция Дамгорда-Меркля с  $n = 256, k = 512$ . Алгоритм использует

- 1) режимы обработки сообщения состоящие из шестидесяти четырех 32-битных слов  $W_0, W_1, \dots, W_{63}$ ,
- 2) восемь рабочих переменных  $a, b, c, d, e, f, g$  и  $h$  из 32 бит каждая и
- 3) хеш-значения из восьми 32-битных слов

$$H^{(i)} = (H_0^{(i)}, H_1^{(i)}, \dots, H_7^{(i)}). \quad (3.1)$$

Конечный результат работы SHA-256 - "отпечаток" сообщения  $M$ , записывается 256 битами. Начальное хеш-значение состоит из восьми слов  $H_0^{(0)}, H_1^{(0)}, H_2^{(0)}, H_3^{(0)}, H_4^{(0)}, H_5^{(0)}, H_6^{(0)}, H_7^{(0)}$ , полученных взятием первых тридцати двух битов дробных частей квадратных корней из первых восьми простых чисел.

Алгоритм хеширования SHA-256 начинается с предварительной подготовки сообщения.

**1. Дополнение сообщения.** Цель этого этапа - добиться, чтобы длина дополненного сообщения была кратна 512 битам. Предположим, что длина сообщения  $M$  равна  $\ell$  битов. Добавим в конце сообщения бит 1 и за ним  $k$  нулей, где  $k$  - наименьшее неотрицательное целое, удовлетворяющее сравнению  $\ell + k + 1 \equiv 448 \pmod{512}$ . Затем добавим ещё блок из 64 битов, равный числу  $\ell$ , записанному в двоичной системе исчисления. После этого длина дополненного сообщения станет кратной 512. Обозначим её  $512 \cdot N$ .

**2. Разбиение сообщения.** Сообщение и его дополнение разбиваются на битовые блоки  $M^{(1)}, M^{(2)}, \dots, M^{(N)}$ , каждый длиной в 512 битов. Так

как эти 512 битов могут быть представлены в виде шестнадцати слов длиной в 32 бита, то первые 32 бита  $i$ -го блока сообщения обозначаются символом  $M_0^{(i)}$ , следующие 32 бита - символом  $M_1^{(i)}$  и так далее до последнего слова  $M_{15}^{(i)}$ .

### 3.2.2. Логические операции и функции.

Напомним некоторые понятия и определения операций, используемых в дальнейшем.

Бит — это минимальная единица измерения объёма информации, так как она хранит одно из двух значений — 0 или 1. То есть одна битовая ячейка может находиться одновременно лишь в одном состоянии из возможных двух - 0 или 1. Есть определённые операции, для манипуляций с битами. Эти операции называются логическими или булевыми операциями, названные в честь математика Джорджа Буля (1815-1864). Все эти операции могут быть применены к любому биту, независимо от того, какое он имеет значение — 0 или 1. Ниже приведены основные логические операции.

$\wedge$  - Логическая операция И (AND) выполняется с двумя битами, назовем их  $a$  и  $b$ . Результат выполнения логической операции И будет равен 1, если  $a$  и  $b$  равны 1, а во всех остальных (других) случаях, результат будет равен 0.

$\vee$  - Логическая операция ИЛИ (OR) выполняется с двумя битами ( $a$  и  $b$ ). Результат выполнения логической операции ИЛИ будет равен 0, если  $a$  и  $b$  равны 0, а во всех остальных случаях, результат равен 1.

$\oplus$  - Логическая операция исключающее ИЛИ (XOR) выполняется с двумя битами ( $a$  и  $b$ ). Результат выполнения логической операции XOR будет равен 1, если один из битов  $a$  или  $b$  равен 1, во всех остальных случаях, результат равен 0 (нулю). Эту операцию иногда называют сложением по модулю 2.

$\neg$  - Логическая операция НЕ (NOT, отрицание) выполняется с одним битом. Результат выполнения этой логической операции напрямую зависит от состояния бита. Если бит находился в нулевом состоянии, то результат выполнения NOT будет равен единице и наоборот.

$+$  - Сложение по модулю  $2^{32}$ .

$\ll$  - Операция сдвига влево, где  $x \ll n$  получается путем отбрасывания крайних левых  $n$  битов слова  $x$ , а затем дополнения результата  $n$

нулями справа.

$\gg$  - Операция сдвига вправо, где  $x \gg n$  получается путем отбрасывания крайних правых  $n$  битов слова  $x$  и последующего дополнения результата  $n$  нулями слева.

$ROTL^n(x)$  - Операция поворота влево (круговое смещение влево), где  $x$  - это 32-битное слово, а  $n$  - целое число с  $0 \leq n < 32$ , определяется как  $ROTL^n(x) = (x \ll n) \vee (x \gg 32 - n)$ .

$ROTR^n(x)$  - Операция поворота вправо (круговое смещение вправо), где  $x$  - это 32 битное слово, а  $n$  - целое число с  $0 \leq n < 32$ , определяется как  $ROTR^n(x) = (x \gg n) \vee (x \ll 32 - n)$ .

$SHR^n(x)$  - Операция сдвига вправо, где  $x$  - это 32 - битное слово, а  $n$  - целое число с  $0 \leq n < 32$ , определяется как  $SHR^n(x) = x \gg n$ .

Алгоритм SHA-256 использует шесть логических функций, где каждая функция работает с 32-битными словами, которые представлены как  $x, y$  и  $z$ . Результатом действия каждой функции является новое 32-битное слово.

$$\begin{aligned} Ch(x, y, z) &= (x \wedge y) \oplus (\neg x \wedge z), \\ Maj(x, y, z) &= (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z), \\ \sum_0(x) &= ROTR^2(x) \oplus ROTR^{13}(x) \oplus ROTR^{22}(x), \\ \sum_1(x) &= ROTR^6(x) \oplus ROTR^{11}(x) \oplus ROTR^{25}(x), \\ \sigma_0(x) &= ROTR^7(x) \oplus ROTR^{18}(x) \oplus SHR^3(x), \\ \sigma_1(x) &= ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x). \end{aligned}$$

Здесь Maj (majority) - обозначает большинство: для каждого индекса бита этот результирующий бит соответствует большинству из 3 входных битов для  $x, y$  и  $z$  в этом индексе. Ch происходит от Choice (выбор). Для каждого индекса результирующий бит соответствует биту из  $y$  (или соответственно  $z$ ) в этом индексе, в зависимости от того, равен ли бит из  $x$  в этом индексе 1 (или, соответственно, 0).

SHA-256 использует последовательность из шестидесяти четырех постоянных 32-разрядных слов:  $K_0, K_1, \dots, K_{63}$ . Эти слова представляют собой первые тридцать два бита дробных частей кубических корней из первых шестидесяти четырех простых чисел.



### 3.2.3. Предварительная подготовка.

Предварительная подготовка состоит из трех этапов:

1. **Дополнение сообщения.** Цель этого этапа - добиться, чтобы длина дополненного сообщения была кратна 512 битам. Предположим, что длина сообщения  $M$  равна  $\ell$  битов. Добавим в конце сообщения бит 1 и за ним  $k$  нулей, где  $k$  - наименьшее неотрицательное целое, удовлетворяющее сравнению  $\ell + k + 1 \equiv 448 \pmod{512}$ . Затем добавим ещё блок из 64 битов, равный числу  $\ell$ , записанному в двоичной системе исчисления. После этого длина дополненного сообщения станет кратной 512. Будем считать, что длина дополненного сообщения равна  $512N$ .

2. **Разбиение сообщения.** Сообщение и его дополнение разбиваются на  $N$  512-битовых блоков  $M^{(1)}, M^{(2)}, \dots, M^{(N)}$ . Так как 512 битов входного блока могут быть представлены в виде шестнадцати слов длиной в 32 битов, то первые 32 бита  $i$ -го блока сообщения обозначаются символом  $M_0^{(i)}$ , следующие 32 бита - символом  $M_1^{(i)}$  и так далее до последнего слова  $M_{15}^{(i)}$ .

3. **Установка начального хеш-значения.** Этот вектор из восьми 32-битных слов  $H_0^{(0)}, H_1^{(0)}, H_2^{(0)}, H_3^{(0)}, H_4^{(0)}, H_5^{(0)}, H_6^{(0)}, H_7^{(0)}$  уже описан ранее.

### 3.2.4. Сжимающая функция для SHA-256.

Алгоритм SHA-256 может использоваться для хеширования любого сообщения  $M$ , имеющего длину меньшую, чем  $2^{64}$  бита. Алгоритм использует

- 1) режимы обработки сообщения из шестидесяти четырех 32-битных слов,
- 2) восемь рабочих переменных из 32 бит каждая и
- 3) хеш-значение из восьми 32-битных слов.

Конечный результат работы SHA-256 - "отпечаток" сообщения, записывается 256 битами. Слова режимов обработки сообщения помечаются буквами  $W_0, W_1, \dots, W_{63}$ . Восемь рабочих переменных помечены как  $a, b, c, d, e, f, g$  и  $h$ . Слова значения хеша помечены символами

$$H^{(i)} = (H_0^{(i)}, H_1^{(i)}, \dots, H_7^{(i)}). \quad (3.2)$$

При  $i = 0$  этот вектор совпадает с начальным хеш-значением, а затем он заменяется каждым последующим промежуточным хеш-значением (3.2) (после обработки каждого блока сообщения), и заканчивая конечным значением хеша,  $H^{(N)}$ , SHA-256 также использует два временных слова,  $T_1$  и  $T_2$ .

При вычислении значения хеш-функции SHA-256 каждый блок сообщения,  $M^{(1)}, M^{(2)}, \dots, M^{(N)}$ , обрабатывается по порядку, на следующих шагах.

От  $i = 1$  до  $N$  выполнить следующие пункты 1-4:

1. Установить слова режимов обработки сообщения

$$W_t = \begin{cases} M_t^{(i)}, & 0 \leq t \leq 15; \\ \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16}, & 16 \leq t \leq 63. \end{cases}$$

2. Инициализировать рабочие переменные  $a, b, c, d, e, f, g$  и  $h$ , полагая их равными восьми 32-битовым словам  $H_j^{(i-1)}$ ,  $0 \leq j \leq 7$ , а именно

$$\begin{aligned} a &= H_0^{(i-1)}, & b &= H_1^{(i-1)}, & c &= H_2^{(i-1)}, & d &= H_3^{(i-1)}, \\ e &= H_4^{(i-1)}, & f &= H_5^{(i-1)}, & g &= H_6^{(i-1)}, & h &= H_7^{(i-1)}. \end{aligned}$$

3. От  $t = 0$  до  $t = 63$  выполнить указанные в этом пункте операции

$$\begin{aligned} T_1 &= h + \Sigma_1(a) + Ch(e, f, g) + K_t + W_t, \\ T_2 &= \Sigma_0(a) + Maj(a, b, c), \\ h &= g, & g &= f, & f &= e, & e &= d + T_1, \\ d &= c, & c &= b, & b &= a, & a &= T_1 + T_2. \end{aligned}$$

4. Вычислить промежуточное хеш-значение с номером  $i$

$$\begin{aligned} H_0^{(i)} &= a + H_0^{(i-1)}, & H_1^{(i)} &= b + H_1^{(i-1)}, & H_2^{(i)} &= c + H_2^{(i-1)}, \\ H_3^{(i)} &= d + H_3^{(i-1)}, & H_4^{(i)} &= e + H_4^{(i-1)}, & H_5^{(i)} &= f + H_5^{(i-1)}, \\ & & H_6^{(i)} &= g + H_6^{(i-1)}, & H_7^{(i)} &= h + H_7^{(i-1)}. \end{aligned}$$

После повторения шагов с первого по четвёртый в совокупности  $N$  раз, т.е. после обработки блока  $M^{(N)}$ , получится 256-битный отпечаток сообщения  $M$ , а именно

$$h(M) = H_0^{(N)} || H_1^{(N)} || H_2^{(N)} || H_3^{(N)} || H_4^{(N)} || H_5^{(N)} || H_6^{(N)} || H_7^{(N)}.$$

**Конец десятой лекции.**