24 Генерация простых

24.1 Генерация простых

Вернемся к языку $PRIMES = \{n \in \{0,1\}^* : n$ — простое$\}$. При генерации параметров RSA нам требовалось вырабатывать случайные простые числа $n \in PRIMES$ заданной битовой длины k: $2^{k-1} \leqslant n < 2^k$.

Генерация простых выполняется по следующей общей схеме:

1. *Некоторым* образом выбрать натуральное n нужной длины.
2. Используя *некоторую* машину M, которая распознает язык $PRIMES$, вычислить ответ $M(n)$.
3. Если $M(n) = 1$, то возвратить n. Иначе — вернуться к шагу 1.

На практике используются два основных варианта уточнения данной схемы.

A. Проверка простоты. На шаге 1 выбирать случайное нечетное n нужной длины. На шаге 2 использовать ПВМТ M, которая дает *всегда* правильные ответы на входе $n \in PRIMES$ и дает *возможно* правильные ответы на входе $n \notin PRIMES$ (алгоритм типа Монте — Карло):

Число	Вероятность ответа	
	простое	составное
простое	1	0
составное	α	$1-\alpha$

Если вероятность ошибки α велика, то можно обратиться к M d раз и понизить ее до α^d.

Алгоритм Проверка простоты

Вход: $k \in \{2,3,\ldots\}$.

Выход: n — вероятно простое число длины k.

Шаги:

1. $n \xleftarrow{R} \{2^{k-1}+1, 2^{k-1}+3, \ldots, 2^k-1\}$.
2. Для $i = 1, \ldots, d$:
   (1) если $M(n) = 0$, то вернуться к шагу 1.
3. Возвратить n.

B. Построение простых. На шаге 1 построить n специальным образом, например, выбрать в качестве n число Мерсенна. На шаге 2 использовать детерминированную машину M, которая распознает такие специальные n за полиномиальное время.

Опишем способ реализации каждого из вариантов.

24.2 Распределение простых

Пусть β — вероятность того, что случайное число из множества $\{2^{k-1}+1, 2^{k-1}+3, \ldots, 2^k-1\}$ является простым. Тогда среднее число кандидатов, которое потребуется проверить для получения простого числа есть

$$\begin{aligned}\sum_{t=1}^{\infty} t\mathbf{P}\{\text{потребуется } t \text{ кандидатов}\} &= \sum_{t=0}^{\infty} \mathbf{P}\{\text{потребуется} > t \text{ кандидатов}\} \\ &= \sum_{t=0}^{\infty} (1-\beta)^t \\ &= \frac{1}{1-(1-\beta)} = \frac{1}{\beta}.\end{aligned}$$

Оценим β. Пусть $\pi(x)$ — число простых чисел $\leqslant x$. Знаменитая теорема аналитической теории чисел гласит, что

$$\pi(x) = \frac{x}{\ln x}(1 + o(1)), \quad x \to \infty.$$

Поэтому среди элементов множества $\{2^{k-1} + 1, 2^{k-1} + 3, \ldots, 2^k - 1\}$ имеется примерно

$$\pi(2^k) - \pi(2^{k-1}) \approx \frac{2^k}{\ln 2^k} - \frac{2^{k-1}}{\ln 2^{k-1}} = \frac{2^{k-1}(k-2)}{k(k-1)\ln 2}$$

простых и вероятность β близка к

$$\frac{\pi(2^k) - \pi(2^{k-1})}{2^{k-2}} \approx \frac{2(k-2)}{k(k-1)\ln 2} \approx \frac{2.88}{k}.$$

Например, при $k = 512$ вероятность $\beta \approx 2.88/512 \approx 1/177$ и для получения простого числа потребуется проверить около 177 кандидатов в среднем.

Остается научиться распознавать простоту чисел.

24.3 Проверка простоты: тест Ферма

Определение 24.1. Число n называется *псевдопростым по основанию* a, если $a^{n-1} \equiv 1 \pmod{n}$. □

Согласно малой теореме Ферма, если n — простое, то n — псевдопростое по любому основанию $a \in \{1, 2, \ldots, n-1\}$ (обратное, вообще говоря, неверно). Данный факт позволяет предложить следующий алгоритм проверки простоты:

Алгоритм Тест Ферма

Вход: n.

Выход: 1 (n — простое) или 0 (n — составное).

Шаги:

1. $a \xleftarrow{R} \{1, 2, 3, \ldots, n-1\}$.
2. Если $(a, n) \neq 1$, то возвратить 0.
3. Если $a^{n-1} \not\equiv 1 \pmod{n}$, то возвратить 0.
4. Возвратить 1.

Обсудим работу алгоритма на составном входе n. Вероятность ошибки

$$\alpha = \mathsf{P}\{(a, n) = 1\}\,\mathsf{P}\left\{a^{n-1} \equiv 1 \pmod{n} \mid (a, n) = 1\right\} = \frac{\varphi(n)}{n-1}\mathsf{P}\left\{a^{n-1} \equiv 1 \pmod{n} \mid (a, n) = 1\right\}.$$

Вероятность $\varphi(n)/(n-1)$ может быть большой, а условная вероятность может равняться 1. Последнее выполняется для, так называемых, *чисел Кармайкла*: n — число Кармайкла, если

$$a^{n-1} \equiv 1 \pmod{n} \quad \forall a \in \mathbb{Z}_n^*.$$

Следовательно, тест Ферма может давать большую ошибку распознавания α и поэтому он не часто используется на практике.

Упражнение 24.1. Первое число Кармайкла: $n = 561 = 3 \cdot 11 \cdot 17$. Найти вероятность $\alpha = \varphi(n)/(n-1)$. □

24.4 Проверка простоты: тест Рабина — Миллера

Определение 24.2. Пусть $n = 2^s r + 1$, где r — нечетное. Число n называется *сильно псевдопростым по основанию* a, если $a^r \equiv 1 \pmod n$ или $a^{2^t r} \equiv -1 \pmod n$ для некоторого $t \in \{0, 1, \ldots, s-1\}$. □

Если n — простое, то по малой теореме Ферма хотя бы одна из скобок в произведении

$$(a^r - 1)(a^r + 1)(a^{2r} + 1) \ldots (a^{2^{s-1}r} + 1) = a^{n-1} - 1$$

делится на n для любого целого a, не кратного n. Поэтому всякое простое является сильно псевдопростым по любому основанию $a \in \{1, 2, \ldots, n-1\}$.

На этом наблюдении построен следующий алгоритм проверки простоты:

Алгоритм Тест Рабина — Миллера

Вход: $n = 2^s r + 1$, r — нечетное.

Выход: 1 (n — простое) или 0 (n — составное).

Шаги:

1. $a \overset{R}{\leftarrow} \{1, 2, 3, \ldots, n-1\}$.
2. Если $(a, n) \neq 1$, то возвратить 0.
3. $v \leftarrow a^r \bmod n$.
4. Если $v \equiv 1 \pmod n$, то возвратить 1.
5. Для $i = 0, \ldots, s-1$:
   (1) если $v \equiv -1 \pmod n$, то возвратить 1;
   (2) $v \leftarrow v^2 \bmod n$.
6. Возвратить 0.

На составном входе n вероятность ошибки

$$\alpha = \mathsf{P}\{(a, n) = 1\}\, \mathsf{P}\{n \text{ – сильно псевдопростое по основанию } a \mid (a, n) = 1\}.$$

Теорема 24.1 (Рабин). Для всякого составного n справедлива оценка

$$\mathsf{P}\{n \text{ – сильно псевдопростое по основанию } a \mid (a, n) = 1\} \leqslant 1/4.$$

Теорема 24.2 (Миллер). Если верна расширенная гипотеза Римана и n является сильно псевдопростым по всем основаниям a из интервала $1 < a < 2\ln^2 n$, то n — простое.

Теорема Рабина означает, что $PRIMES \in \mathbf{BPP}$, из теоремы Миллера следует, что $PRIMES \in \mathbf{P}$ при справедливости расширенной гипотезы Римана.

Упражнение 24.2. Доказать, что всякое сильно псевдопростое по основанию a является псевдопростым по этому основанию. □

24.5 Построение простых

В отечественном стандарте `СТБ 1176.2-99` (и в прекратившем действие российском стандарте `ГОСТ Р 34.10-94`) простые строятся на основании следующего утверждения.

Теорема 24.3 (Диемитко). Пусть $n = qR + 1$, где q — нечетное простое, R — четное и $R < 4(q+1)$, т. е. $n < (2q+1)^2$. Если

(1) $2^{qR} \equiv 1 \pmod{n}$,

(2) $2^R \not\equiv 1 \pmod{n}$,

то n — простое.

Доказательство. Пусть $n = p_1^{e_1} \dots p_k^{e_k}$. И пусть d — порядок числа 2 по модулю n, т. е. d — минимальное натуральное такое, что $2^d \equiv 1 \pmod{n}$.

Тогда

а) $d \mid qR$ в силу (1);

б) $d \not| R$ в силу (2);

в) $d \mid \varphi(n)$ по теореме Эйлера.

Из а) и б) следует, что г) $q \mid d$, а из в) и г) следует, что

$$q \mid \varphi(n) = p_1^{e_1-1} \dots p_s^{e_k-1}(p_1 - 1) \cdots (p_k - 1).$$

Предположим, что q совпадает с одним из p_i. Тогда существует натуральное r такое, что $n = qr$, и, значит, в силу условия теоремы $qr = qR + 1$, противоречие.

Таким образом, q должен делить один из множителей $p_i - 1$, т. е. $p_i = qr + 1$ для некоторого r. Отсюда $n = p_i m = (qr + 1)m = qR + 1$ и, следовательно, $m = q(R - rm) + 1$. Итак,

$$n = (qr + 1)(qs + 1),$$

где r и $s = R - rm$ — четные числа, причем $r \geqslant 2$, $s \geqslant 0$.

Предположим теперь, что n — составное. Тогда $s \geqslant 2$, откуда следует $n \geqslant (2q+1)^2$. Из этого противоречия следует, что $s = 0$ и, значит, $n = p_i$ — простое. □

Алгоритм Построение простых

Вход: $l \in \{2, 3, \dots\}$.

Выход: p — простое число длины ℓ.

Шаги:

1. Построить последовательность $\ell_0, \ell_1, \dots, \ell_s$, в которой $\ell_0 = \ell$, $\ell_{i+1} = \lfloor (\ell_i + 1)/2 \rfloor$, $i = 0, 1, \dots, s - 1$, и $2 \leqslant \ell_s \leqslant \Delta$. Здесь Δ мало настолько, что можно без труда находить простые длины $\leqslant \Delta$.

2. Выбрать простое p_s длины ℓ_s.

3. Для $i = s - 1, \dots, 1, 0$:

   (1) выработать случайное четное R такое, что $p_{i+1}R + 1$ является ℓ_i-битовым;

   (2) установить $p_i \leftarrow p_{i+1}R + 1$;

   (3) проверить условия $2^{p_i-1} \equiv 1 \pmod{p_i}$, $2^R \not\equiv 1 \pmod{p_i}$;

   (4) если не выполняется хотя бы одно из условий, то вернуться к шагу 3.1.

4. Возвратить p_0.

Корректность. Для числа R, которое определяется на шаге 3.1, выполняется ограничение $R < 4(p_{i+1} + 1)$ и можно применять теорему Диемитко. Действительно, пусть, от противного, $R \geqslant 4(p_{i+1} + 1)$. Число p_{i+1} является ℓ_{i+1}-битовым, поэтому

$$p_{i+1} > 2^{\ell_{i+1}-1} \Rightarrow R > 2^{\ell_{i+1}+1} \Rightarrow p_i = p_{i+1}R + 1 > 2^{2\ell_{i+1}}$$

и битовая длина p_i не меньше $2\ell_{i+1} + 1$, т. е. $\ell_i \geqslant 2\ell_{i+1} + 1$. Но это противоречит выбору $\ell_{i+1} = \lfloor (\ell_i + 1)/2 \rfloor$.
