ТЕОРЕТИКО-ЧИСЛОВЫЕ МЕТОДЫ В КРИПТОГРАФИИ

Лекция 14

Факторизация, простота чисел и дискретный логарифм

Тема 14 по рабочей программе: факторизация чисел; тесты на простоту; генерация простых чисел; генератор и дискретный логарифм. **Аудиторная нагрузка:** 1 академический час. Конспект содержит расширенные объяснения для самостоятельного изучения.

Цель лекции: связать алгоритмы теории чисел с построением и проверкой параметров криптосистем; различать найденный делитель, свидетельство составности и доказательство простоты.

После изучения темы студент должен уметь: применять метод Ферма и ро-метод Полларда к небольшим числам; выполнять раунд Миллера — Рабина; проверять сертификат простоты; находить образующий элемент и дискретный логарифм методом больших и малых шагов.

Предварительная подготовка: алгоритм Евклида, сравнения и модульное возведение в степень, малая теорема Ферма, порядок элемента, циклическая группа, разложение целого числа на простые множители.

План лекции

1. Факторизация и проверка простоты как разные задачи.
2. Метод Ферма и ро-метод Полларда.
3. Тест Миллера — Рабина и границы вероятностного вывода.
4. Построение простых чисел и сертификат на основе факторизации числа, уменьшенного на единицу.
5. Образующий элемент и метод больших и малых шагов.

Для занятия продолжительностью 45 минут: постановка задач — 4 минуты; факторизация — 10; проверка простоты — 9; генерация и сертификат — 10; образующий элемент и дискретный логарифм — 10; итоговые вопросы — 2. Полные доказательства и упражнения изучаются также самостоятельно.

1 Какие задачи необходимо различать

Факторизация целого числа $N > 1$ — его разложение на простые множители. Промежуточная задача состоит в нахождении нетривиального делителя d, для которого $1 < d < N$ и $d \mid N$. Найденные множители могут оказаться составными: тогда их требуется разложить дальше. Проверить предложенное разложение обычно значительно проще, чем найти его.

Проверка простоты отвечает на вопрос, имеет ли число ровно два положительных делителя. Свидетельство составности может доказывать наличие делителей, не предъявляя сами делители. Поэтому отрицательный результат теста простоты ещё не является алгоритмом полной факторизации.

Следует различать три результата: «число составное»; «число прошло перечисленные вероятностные проверки»; «простота доказана». Последний вывод требует точного

алгоритма либо проверенного сертификата. Истечение времени работы алгоритма не заменяет ни один из этих результатов.

Длина входа для целого числа равна числу его двоичных разрядов:

$$L = \lfloor \log_2 N \rfloor + 1.$$

Например, перебор порядка $\sqrt{N}$ делителей экспоненциален по длине L, поскольку $\sqrt{N}$ имеет порядок $2^{L/2}$. Далее оценки в числе арифметических или групповых операций отделяются от битовой стоимости этих операций.

2 Факторизация чисел

2.1 Пробные деления

Если составное число $N = uv$, где $1 < u \le v$, то $u^2 \le N$. Следовательно, у составного числа имеется простой делитель не больше $\sqrt{N}$. Достаточно проверить делимость на 2, затем на нечётные числа до целой части квадратного корня. Проверка составных делителей избыточна, но не нарушает корректность.

Для полной факторизации каждый найденный делитель последовательно выделяют, уменьшая текущее число. Если после перебора остаток больше единицы, он прост. Метод удобен для учебных чисел, проверки найденных множителей и факторизации небольшого числа $p - 1$.

2.2 Метод Ферма: разность квадратов

Для нечётного N ищут целые $a \ge b \ge 0$, удовлетворяющие равенству

$$N = a^2 - b^2 = (a - b)(a + b).$$

Начинают с $a_0 = \lceil \sqrt{N} \rceil$. Для каждого a вычисляют $D = a^2 - N$ и проверяют, является ли D полным квадратом. Проверка должна быть точной: вычисляют $b = \lfloor \sqrt{D} \rfloor$ целочисленным алгоритмом и сравнивают b^2 с D.

Пример 1. Пусть $N = 5959$. При $a = 78$ получаем $D = 125$, при $a = 79$ — $D = 282$, при $a = 80$ — $D = 441 = 21^2$. Следовательно,

$$5959 = (80 - 21)(80 + 21) = 59 \cdot 101.$$

Выполнены три проверки значений D и два увеличения переменной a.

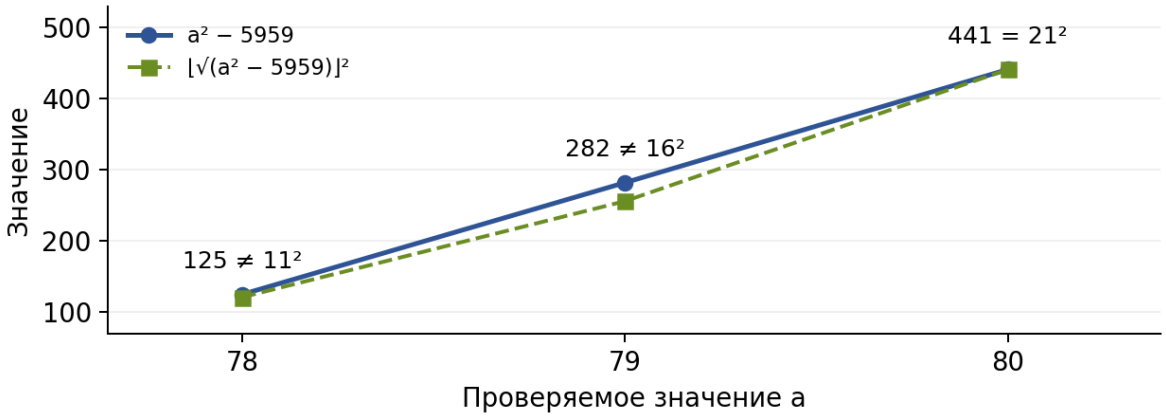


Рисунок 1. Метод Ферма: совпадение с полным квадратом при третьей проверке.

Если $N = pq$, где $p \le q$ — нечётные простые числа, успешное значение равно $a_* = (p+q)/2$. Число проверок в описанной реализации составляет

$$T = a_* - \lceil \sqrt{N} \rceil + 1.$$

Близость множителей объясняет быстроту метода:

$$a_* - \sqrt{N} = \frac{\left(\sqrt{q} - \sqrt{p}\right)^2}{2} = \frac{(q-p)^2}{2\left(\sqrt{p} + \sqrt{q}\right)^2}.$$

При близких p и q последнее выражение приближённо равно $(q-p)^2/\left(8\sqrt{N}\right)$. Это оценка расстояния до успешного значения, а не точное число проверок. Для $N = 8051 = 83 \cdot 97$ уже начальное $a = 90$ даёт $D = 49$.

Метод применим и к нечётному квадрату: при $N = 49$ первая проверка даёт $a = 7$, $b = 0$. Для нечётного простого числа получится только тривиальная пара 1 и N. Ограниченный по числу шагов поиск сам по себе не доказывает простоту.

2.3 Ро-метод Полларда

Пусть у N имеется неизвестный простой делитель p. Последовательность, вычисляемая по правилу $f(x) = x^2 + c \pmod N$, одновременно задаёт последовательность по модулю p: сравнимые числа после применения многочлена остаются сравнимыми. Совпадение двух членов по модулю p означает, что их разность делится на p.

Хранить все члены не обязательно. Используется схема Флойда: одна переменная проходит один шаг, другая — два. При начальных $x = y = x_0$ выполняют

$$x \leftarrow f(x), \qquad y \leftarrow f(f(y)), \qquad d = \gcd(|x-y|, N).$$

При $d = 1$ поиск продолжается. При $1 < d < N$ делитель найден. При $d = N$ данная траектория не дала нетривиального делителя; следует изменить c или начальное значение и начать заново. Ограничение числа итераций также может потребовать нового запуска.

Пример 2. Для $N = 8051$, $c = 1$, $x_0 = 2$ первые строки вычислений имеют следующий вид.

Итерация	x	y	d
1	5	26	1
2	26	7474	1
3	677	871	97

На третьей итерации найден делитель 97, а второй множитель равен 83. Непосредственная проверка даёт $97 \cdot 83 = 8051$.

Название метода связано с формой траектории в конечном множестве: сначала идут неповторяющиеся значения, затем начинается цикл. При эвристическом предположении о поведении последовательности как случайного отображения столкновение по модулю p ожидается примерно через $\sqrt{p}$ шагов. Для произведения двух близких простых это даёт порядок $N^{1/4}$ итераций. Это не гарантированная оценка худшего случая для фиксированного многочлена.

Сама схема Флойда хранит постоянное число целых значений. Таблица всех итераций в методичке добавлена для обучения и требует памяти, пропорциональной числу записанных строк.

3 Вероятностная проверка простоты

3.1 Один раунд Миллера — Рабина

Числа меньше 2, число 2 и чётные числа обрабатываются отдельно. Для нечётного $n \geq 5$ представим

$$n - 1 = 2^s d, \qquad d \text{ нечётно}, \qquad s \geq 1.$$

Выбирают основание a из диапазона от 2 до $n - 2$. Если $\gcd(a, n) > 1$, найдено свидетельство составности и одновременно делитель. Иначе вычисляют $x_0 = a^d \bmod n$ и при необходимости последующие квадраты:

$$x_{j+1} = x_j^2 \bmod n, \qquad 0 \leq j < s - 1.$$

Раунд пройден, если $x_0 = 1$ либо хотя бы один из остатков $x_0, \ldots, x_{s-1}$ равен $n - 1$. Если это не выполнено, число составное. После появления 1 при возведении в квадрат, когда раньше не было -1, можно немедленно завершить раунд с отрицательным результатом.

Почему простое число проходит раунд. Для простого n и ненулевого основания теорема Ферма даёт $a^{n-1} = 1$ в поле $\mathbb{F}_n$. В цепочке последовательных квадратов, продолженной до этого значения, либо первый элемент равен 1, либо перед первым появлением 1 стоит корень уравнения $z^2 = 1$, отличный от 1. В поле такими корнями являются только 1 и -1, поэтому необходимое значение -1 имеется среди проверяемых остатков.

3.2 Что означает положительный результат

Пример 3. Для $n = 2047 = 23 \cdot 89$ имеем $s = 1$, $d = 1023$. При основании 2 получаем $2^{1023} \equiv 1 \pmod{2047}$, и раунд пройден. При основании 3 получаем остаток 1565, отличный от 1 и 2046; число признаётся составным.

Составное число, прошедшее раунд при взаимно простом основании, называют сильным псевдопростым по этому основанию. Один положительный результат не является доказательством простоты. Два заранее фиксированных основания также не обеспечивают доказательство для произвольного размера входа.

Для каждого фиксированного нечётного составного числа доля оснований, при которых раунд ошибочно принимает его, не превосходит одной четверти. Поэтому для k независимых равномерных случайных оснований

$$\Pr(\text{все раунды пройдены} \mid n \text{ фиксировано и составное}) \leq 4^{-k}.$$

Здесь оценивается вероятность поведения теста на заранее фиксированном составном числе. Это не автоматически вероятность составности числа, выбранного генератором после серии проверок: для такого вывода требуется учитывать процедуру выбора кандидатов. Основания 2 и 3 в практикуме выбраны для воспроизводимости вычислений, поэтому приписывать им оценку 4^{-2} нельзя.

Одно возведение в степень с показателем длины порядка L требует порядка L модульных умножений методом повторного возведения в квадрат. Битовая стоимость раунда дополнительно зависит от алгоритмов умножения и редукции. Подробности этих операций и числа Кармайкла рассматриваются в следующей теме.

4 Генерация и доказательство простоты

4.1 Выбор кандидатов

В учебном задании удобно строить наименьшее простое $p \geq A$: переходить по нечётным кандидатам и применять точную проверку пробными делениями. Такой способ воспроизводим, но не создаёт равномерно случайного простого числа.

Для случайного выбора среди b-битовых простых можно независимо генерировать нечётные b-битовые кандидаты и отбрасывать составные. Старший бит фиксирует длину, младший — нечётность. При $b \geq 3$ имеется 2^{b-2} возможных кандидатов. Если кандидаты равномерны, а проверка точна, принятые простые также равномерны среди простых данного диапазона. Это следует из одинаковой вероятности предложить каждый допустимый кандидат на любом шаге.

По теореме о распределении простых их плотность около большого числа x приближённо равна $1/\ln x$. Среди нечётных b-битовых кандидатов вероятность простоты примерно равна $2/(b\ln 2)$, а среднее число попыток — $b\ln 2/2$. Это асимптотические ориентиры, а не гарантированное число шагов.

Практические процедуры построения больших простых сочетают случайный выбор кандидатов, предварительное отсеивание и предусмотренные процедурой тесты либо доказательства. Например, FIPS 186-5 отдельно описывает получение доказанно простых и вероятно простых параметров RSA, а также вероятностные проверки [5]. Функции практикума ограничены малыми числами и предназначены для разбора алгоритмов.

4.2 Сертификат на основе полной факторизации числа n − 1

Пусть $n > 2$ и известно полное разложение $n - 1$ на простые множители. Обозначим различные простые делители через $r_1, \ldots, r_t$.

Критерий простоты. Если существует целое a, для которого

$$a^{n-1} \equiv 1 \pmod{n},$$

$$\gcd(a^{(n-1)/r_i} - 1, n) = 1 \quad \text{для каждого } i,$$

то n простое. В вычислении НОД степень можно предварительно заменить её остатком по модулю n.

Доказательство. Возьмём любой простой делитель q числа n. Первое сравнение гарантирует существование порядка $t = \mathrm{ord}_q(a)$ и делимость $t \mid n - 1$. Условия с НОД означают, что ни для одного r_i порядок t не делит $(n-1)/r_i$. Если бы t был собственным делителем $n - 1$, в его разложении недоставало бы хотя бы одной степени некоторого r_i, и эта делимость выполнялась бы. Значит, $t = n - 1$. По теореме Лагранжа $n - 1 \mid q - 1$, поэтому $q \geq n$. Но $q \mid n$ и $q \leq n$, откуда $q = n$; число n простое.

Пример 4. Для $n = 97$ имеем $96 = 2^5 \cdot 3$. При $a = 5$ выполняются

$$5^{96} \equiv 1, \qquad 5^{48} \equiv 96, \qquad 5^{32} \equiv 35 \pmod{97}.$$

Поскольку $\gcd(95,97) = \gcd(34,97) = 1$, получен сертификат простоты. В него входят число 97, свидетель 5 и полная факторизация 96 с подтверждённой простотой множителей 2 и 3.

Это одна из форм классического критерия Люка по порядку элемента. Её не следует смешивать с вероятностными тестами, использующими последовательности Люка. Нельзя

заменять условия с НОД простыми неравенствами остатков единице, пока простота модуля ещё не доказана. Нахождение полной факторизации $n-1$ само по себе может быть трудной задачей; критерий не устраняет эту трудность.

5 Образующий элемент и дискретный логарифм

5.1 Проверка образующего элемента

Для простого p мультипликативная группа $\mathbb{F}_p^*$ циклическая и имеет порядок $p-1$. Элемент g называется образующим, или первообразным корнем, если его порядок равен $p-1$. Слово «генератор» здесь обозначает элемент группы, а не алгоритм генерации псевдослучайных битов из предыдущей лекции.

Пусть известны все различные простые делители r числа $p-1$. Тогда

$$\operatorname{ord}_p(g) = p-1 \quad \Leftrightarrow \quad g^{(p-1)/r} \not\equiv 1 \pmod{p} \text{ для всех } r.$$

Доказательство повторяет рассуждение о собственном делителе порядка в предыдущем разделе. Здесь модуль уже прост, поэтому любой ненулевой остаток разности со значением 1 автоматически взаимно прост с модулем.

При $p = 97$ элемент $g = 5$ является образующим: остатки 96 и 35 из примера 4 не равны 1. Для нахождения образующего можно проверять кандидаты $2, 3, \ldots$ и выбирать первый подходящий. В практикуме требуется именно наименьший положительный образующий, чтобы ответы были однозначны.

5.2 Задача дискретного логарифмирования

В циклической группе с образующим g порядка n для заданного h требуется найти x из диапазона $0 \le x < n$, удовлетворяющий равенству $g^x = h$. Показатель определён по модулю порядка группы. Полный перебор требует до n последовательных групповых умножений.

Метод больших и малых шагов, или BSGS, использует $m = \lceil \sqrt{n} \rceil$ и представление $x = im + j$, где $0 \le i, j < m$. Из исходного равенства следует

$$g^j = h\,(g^{-m})^i.$$

Сначала сохраняют **малые шаги** $B_j = g^j$ вместе с индексами j. Затем вычисляют **большие шаги** $H_i = h(g^{-m})^i$ и ищут совпадение с сохранённым значением. Если $H_i = B_j$, кандидат равен $x = im + j$. Проверяют диапазон $x < n$ и исходное равенство.

Пример 5. Решим $5^x \equiv 56 \pmod{97}$. Здесь $n = 96$, $m = 10$. Таблица малых шагов содержит остатки

$$1,\ 5,\ 25,\ 28,\ 43,\ 21,\ 8,\ 40,\ 6,\ 30.$$

Множитель большого шага равен $5^{-10} \equiv 11 \pmod{97}$. Значения больших шагов до совпадения: 56, 34, 83, 40. Совпадение соответствует $i = 3$, $j = 7$, откуда $x = 37$. Проверка: $5^{37} \equiv 56 \pmod{97}$.

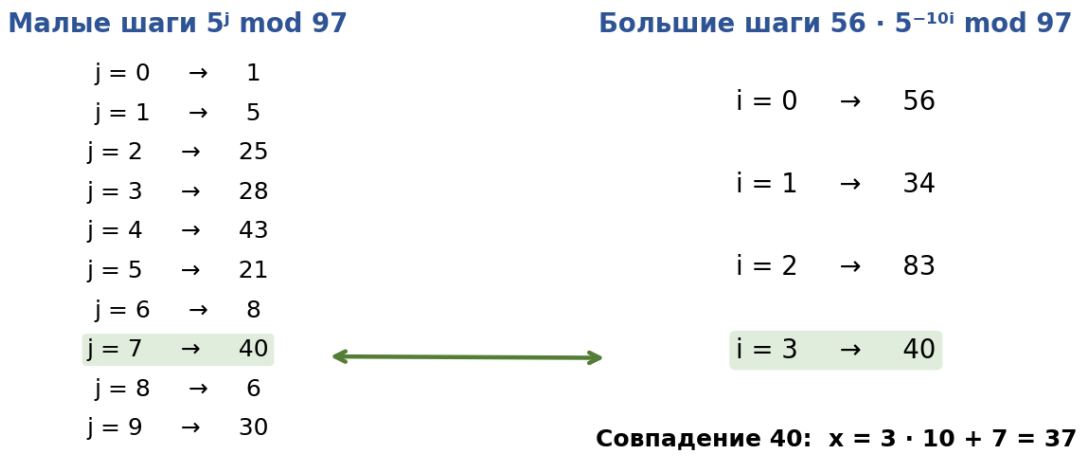


Рисунок 2. Совпадение в двух последовательностях определяет показатель 37.

Алгоритм выполняет порядка $\sqrt{n}$ групповых умножений и хранит порядка $\sqrt{n}$ элементов. Для быстрого поиска совпадений подходит словарь; оценка среднего времени поиска в нём опирается на свойства хеширования. На входе должны быть известны порядок и принадлежность h нужной группе. В практикуме выбран образующий всей группы $\mathbb{F}_p^*$, поэтому любой $h \neq 0$ допустим, а порядок равен $p - 1$ [1, 3].

Сам алгоритм BSGS является общим методом для циклических групп. Он не утверждает, что никакой другой метод с использованием специальной структуры группы не может работать быстрее.

6 Как связаны изученные алгоритмы

Метод	Что получаем	Условие или ограничение
Пробные деления	Точный ответ, простые множители	Для малых чисел; до квадратного корня
Ферма	Два множителя нечётного числа	Эффективен при близких множителях
Ро Полларда	Нетривиальный делитель	Неудачная траектория требует перезапуска
Миллер — Рабин	Составность либо прохождение раунда	Прохождение не доказывает простоту
Сертификат по $n - 1$	Доказательство простоты	Полная факторизация и условия на НОД
Критерий образующего	Полный порядок элемента	Простой модуль, известные делители p − 1
BSGS	Дискретный логарифм	Известный порядок; таблица размера порядка его корня

Факторизация N и факторизация $p - 1$ имеют разные роли: первая может раскрывать множители модуля, вторая помогает доказать простоту кандидата и проверить порядок элемента. Вычисление g^x быстро выполняется модульным возведением в степень, а обратная задача требует отдельного алгоритма.

Упражнения для закрепления

Задача 1. Разложите 5959 методом Ферма. Сколько было проверок разности и сколько увеличений переменной a?

Задача 2. Выполните первые три итерации ро-метода для $N = 8051$, $c = 1$, $x_0 = 2$. Объясните смысл каждого полученного НОД.

Задача 3. Проведите раунды Миллера — Рабина для 2047 по основаниям 2 и 3. Почему положительный результат первого раунда не противоречит разложению $23 \cdot 89$?

Задача 4. Проверьте сертификат простоты числа 101 с $a = 2$ и разложением $100 = 2^2 \cdot 5^2$. Является ли 2 образующим группы ненулевых остатков?

Задача 5. Решите $5^x \equiv 56 \pmod{97}$ методом BSGS. Укажите размер таблицы, индексы совпадения и диапазон ответа. Как изменится ответ при правой части 1?

Задача 6. Докажите, что равномерный выбор нечётного b-битового кандидата с точным отбрасыванием составных даёт равномерное распределение среди b-битовых простых. Объясните, почему последовательный поиск от случайной точки не обладает тем же обоснованием.

Контрольные вопросы

1. Чем свидетельство составности отличается от полной факторизации?
2. Почему перебор до квадратного корня не является полиномиальным по длине двоичной записи?
3. При каких множителях метод Ферма особенно эффективен?
4. Что означают три случая для НОД в ро-методе Полларда?
5. Почему простое число проходит каждый допустимый раунд Миллера — Рабина?
6. Какие условия нужны для оценки вероятности ошибки через число раундов?
7. Чем воспроизводимый поиск следующего простого отличается от случайной генерации?
8. Почему в сертификате важны полнота факторизации и проверка НОД?
9. Как проверить образующий, не перебирая все его степени?
10. Почему BSGS использует таблицу размера порядка квадратного корня из порядка группы?

Литература и источники

1. Boneh D., Shoup V. *A Graduate Course in Applied Cryptography*. Версия 0.6, январь 2023, § 16.1.1, с. 682–684: метод больших и малых шагов. Использован предоставленный файл book.pdf.
2. Katz J., Lindell Y. *Introduction to Modern Cryptography*. 2nd ed. CRC Press, 2015. § 9.1.2, с. 344–345: ро-метод Полларда и эвристическая оценка. Использовано предоставленное второе издание.
3. Trappe W., Washington L. C. *Introduction to Cryptography with Coding Theory*. 2nd ed. Pearson, 2006. § 6.3, с. 176–179; § 6.4, с. 181–182; § 7.2.2, с. 206–207. В § 6.4 перебор для метода Ферма организован иначе; число проверок в настоящем конспекте выведено для последовательного увеличения a.
4. Мартынов Л. М. *Алгебра и теория чисел для криптографии*. 2020. Предоставленное издание: дополнительная русскоязычная литература по делимости, группам и порядкам элементов.
5. NIST. *FIPS 186-5: Digital Signature Standard (DSS)*. 2023. Приложения A.1 и B.3. Официальная публикация. Связь учебных алгоритмов с процедурами формирования параметров электронной подписи.