0.0 Элементы криптографии: математические основы

◇ 1. Сложность алгоритма

- В криптографии важно оценивать, насколько быстро работает алгоритм при увеличении размера входных данных.
- **Сложность алгоритма** — это количество операций (сложения, умножения и т.д.), которые он выполняет.
- Оценивается по длине входа (количеству битов N).

Типы сложности:

- **Полиномиальная**: выражается многочленом от N → считается «быстрой».
- **Субэкспоненциальная**: что-то между полиномиальной и экспоненциальной.
- **Экспоненциальная**: растёт очень быстро → алгоритм считается «медленным» и непрактичным для больших входов.

Пример: разложение числа на множители — классическая задача криптографии. Её решение зависит от сложности выбранного алгоритма.

---

2. Расширенный алгоритм Евклида

Этот алгоритм не только находит наибольший общий делитель (НОД) чисел a и b, но и числа x и y, такие что:

$$ a \cdot x + b \cdot y = \text{НОД}(a, b) $$

Особенно полезен, если нужно найти **обратное число по модулю** (используется в RSA).

Алгоритм работает итеративно, обновляя переменные r_i, x_i, y_i — пока не найдёт НОД и соответствующие коэффициенты.

---

3. Бинарный алгоритм возведения в степень по модулю

Цель: вычислить $a^n \mod m$ быстро.

Обычный способ — слишком медленный: выполняет n-1 умножение.

Бинарный способ:

1. Представь n в двоичном виде.
2. Возводи a в квадрат и умножай по правилам битов.
3. Работает за логарифмическое время (~1.5 log(n) умножений).

Широко используется в криптографии (например, при шифровании в RSA).

---

4. Модульная арифметика

Это арифметика по остатку деления:

- $a \equiv b \ (\text{mod } m)$ ⇔ остатки от деления a и b на m совпадают.
- Свойства (можно складывать, вычитать, умножать и даже делить — если делитель обратим).

Пример: $2^{345} \mod 31 = 1$ (используя правила возведения в степень и модульную арифметику).

---

5. Функция Эйлера ($\varphi(n)$)

$$ \varphi(n) = \text{кол-во чисел } < n, \text{ взаимно простых с } n $$

Примеры:

- $\varphi(5) = 4$ (все числа 1–4 кроме 5 — взаимно просты с 5)
- $\varphi(12) = 4$ (взаимно просты: 1, 5, 7, 11)

Полезна в теореме Эйлера, RSA и при нахождении обратных чисел.

Формула: Если $n = p_1^{a_1} \cdot ... \cdot p_k^{a_k}$, то:

$$ \varphi(n) = n \cdot \left(1 - \frac{1}{p_1}\right) \cdot \dots \cdot \left(1 - \frac{1}{p_k}\right) $$

---

6. Сравнения по модулю и их свойства

- $a \equiv b \mod m$: числа дают один и тот же остаток при делении на m.

- Можно:

  - складывать, вычитать, умножать,
  - возводить в степень,
  - делить (если делитель взаимно прост с модулем).

◇ 7. Малая теорема Ферма

Если p — простое, и $a \not\equiv 0 \mod p$, то:

$$ a^{p-1} \equiv 1 \mod p $$

Основа RSA и других криптографических методов.

◇ 8. Теорема Эйлера (обобщение Ферма)

Если $(a, m) = 1$, то:

$$ a^{\varphi(m)} \equiv 1 \mod m $$

Позволяет быстро вычислять большие степени по модулю.

---

9. Методы решения сравнений первой степени

Пример: $a \cdot x \equiv b \mod m$

Методы:

1. Через **обратное число**: $x \equiv a^{-1} \cdot b \mod m$
2. Через **расширенный алгоритм Евклида**
3. Через **теорему Эйлера**: $a^{-1} = a^{\varphi(m)-1} \mod m$
4. **Метод Ньютона** (для степеней 2^k)
5. В системе **Maple** — с помощью команды `msolve`.

---

10. Мультипликативно обратное число

Число $a^{-1} \mod m$ такое, что:

$$ a \cdot a^{-1} \equiv 1 \mod m $$

Существует, если $(a, m) = 1$.

Используется для деления по модулю.

---

11. Китайская теорема об остатках (КТО)

Если есть система сравнений:

$$ x \equiv a_1 \mod m_1 \ x \equiv a_2 \mod m_2 \ \text{и т.д.} $$

и все m_i попарно взаимно просты — можно найти одно общее решение по формуле.

Позволяет решать системы сравнений, применимо в криптографических схемах (например, оптимизация RSA).

◇ 12. Показатель и первообразный корень по модулю

- **Показатель числа a** по модулю m — наименьшее d, при котором $a^d \equiv 1 \mod m$.
- Если существует такое число a, что его степени покрывают все ненулевые остатки по модулю — оно называется **первообразным корнем**.

◇ 13. Дискретный логарифм

Это задача нахождения x из:

$$ a^x \equiv b \mod m $$

Очень сложна для больших m → используется как основа криптографической стойкости.

◇ 14. Парадокс дней рождения

Говорит о том, что вероятность коллизии (совпадения хэшей и т.п.) гораздо выше, чем ожидается. Применяется в криптоанализе.

УПРАЖНЕНИЕ 0.3. Асимметричная криптография: алгоритм и протокол RSA

Введение в информационную безопасность и криптографию

Основные задачи информационной безопасности:

1. **Сохранение тайны информации**.
2. **Передача информации скрытно от третьих лиц**.

Три подхода к решению:

- Создание **надёжных хранилищ/каналов**.
- Использование **криптографических систем** (преобразование информации).
- **Стеганография** — скрытие самого факта передачи данных.

Классификация криптографических систем

1. Секретные (симметричные) системы

- Один ключ для шифрования и дешифрования.
- Примеры: **AES, RC6, Кузнечик**.
- Проблема: необходим **защищённый обмен ключами**.

2. Системы с открытым ключом (асимметричные)

- Пара ключей:
  - Открытый ключ — доступен всем.
  - Закрытый (секретный) — хранится владельцем.
- Примеры: **RSA, Диффи-Хеллман, Эль-Гамаль**.
- Позволяют использовать **открытые каналы связи**.

Криптографические функции

1. Односторонняя функция

Функция $f(x)$, которую:

- **Легко** вычислить.
- **Сложно** обратить (найти x по y).

> Пример: $y = 3^x \mod 17$ — легко посчитать, но трудно найти x, зная y (дискретное логарифмирование).

2. Односторонняя функция с «лазейкой» (trapdoor function)

- При наличии **секретной информации** (лазейки) обращение функции становится простым.
- Используется в **RSA**, где знание разложения модуля m на простые позволяет эффективно расшифровывать.

Алгоритм RSA

Основные идеи:

- Используется функция $y = x^e \mod m$, где:
  - $m = p \cdot q$ — произведение двух больших простых.
  - e — открытый показатель.
- Расшифровка возможна, если знаешь d, такой что $ed \equiv 1 \mod \varphi(m)$.

Этапы создания ключей:

1. Выбираются два больших простых числа p, q.
2. Считается модуль $m = p \cdot q$.
3. Вычисляется функция Эйлера: $\varphi(m) = (p-1)(q-1)$.
4. Выбирается открытый ключ e, такой что $\gcd(e, \varphi(m)) = 1$.
5. Вычисляется закрытый ключ d, обратный к $e \mod \varphi(m)$.

Пара (e, m) — открытый ключ. Пара (d, m) — закрытый ключ.

Шифрование и расшифровка

- **Шифрование**: $c = m^e \mod m$
- **Расшифровка**: $m = c^d \mod m$

> Благодаря свойствам теоремы Эйлера и модульной арифметики, гарантируется корректное восстановление исходного сообщения.

Ускорение с помощью Китайской теоремы об остатках

При расшифровке RSA можно ускорить процесс, если:

- Вместо одного большого модуля m использовать:
  - $c^d \mod p$
  - $c^d \mod q$
- Затем объединить результаты через **Китайскую теорему об остатках (КТО)**.

Протокол RSA (обмен сообщениями)

1. **Абонент B** создаёт ключи и публикует свой открытый ключ.

2. **Абонент A**, чтобы отправить сообщение m:

   - Шифрует: $c = m^e \mod m$ с помощью открытого ключа B.

3. **B** расшифровывает: $m = c^d \mod m$ с помощью своего закрытого ключа.

Открытые ключи могут находиться в «телефонной книге».

Безопасность RSA

- Надёжность RSA основана на **сложности факторизации больших чисел**.

- Поэтому:

  - Простые числа должны быть **достаточно большими** (100+ цифр).
  - Иногда используют так называемые **сильные простые числа**, чтобы усложнить факторизацию.

Заключение

RSA — основа современной криптографии с открытым ключом. Он:

- Позволяет безопасно обмениваться сообщениями через открытые каналы.
- Основан на твёрдых математических принципах.
- Используется не только для шифрования, но и для **цифровой подписи**.

Отлично! Вот подробный и понятный конспект по **Упражнению 4: Асимметричная криптография – электронная подпись и аутентификация на базе RSA**, подготовленный для зачёта.

Упражнение 0.4. Электронная подпись и аутентификация на базе RSA

---

1. Что такое электронная цифровая подпись (ЭЦП)?

ЭЦП — это реквизит электронного документа, который:

1. Подтверждает, что документ **не был искажён** при передаче.
2. Гарантирует, что его подписал **определённый отправитель**.

ЭЦП = аналог рукописной подписи в цифровом мире.

Виды подписей:

- Рукописная.
- Нотариальная.

- Электронная личная.
- Сертифицированная электронная (аналог нотариальной).

Свойства ЭЦП:

- Только владелец подписи может её поставить.
- Автор не может **отказаться** от подписи.
- Третьи стороны могут **проверить подлинность**.

---

2. Электронная подпись на базе RSA (без хэширования)

Протокол:

1. Отправитель A создаёт пару ключей RSA:
    - открытый ключ: (e, m)
    - закрытый ключ: (d, m)
2. Сообщение x подписывается: $s = x^d \mod m$
3. Получателю B передаётся пара (x, s)
4. Получатель проверяет подлинность: $x' = s^e \mod m$ если $x' = x$, подпись подлинна.

Свойства:

- Подделать подпись невозможно (нужно знать d).
- Невозможно «подставить» другой текст под подпись.
- Это называется **неотказуемостью** (non-repudiation).

---

3. Электронная подпись с хэшированием

Используется, когда сообщение большое — подписывают не весь текст, а его **хэш**.

Протокол:

1. Алиса выбирает параметры RSA:
    - два простых числа P, Q
    - $N = P \cdot Q$, $\varphi(N) = (P - 1)(Q - 1)$
    - закрытый ключ e, открытый $d = e^{-1} \mod \varphi(N)$
2. Алиса хочет подписать сообщение m:
    - Вычисляет хэш: $y = h(m)$
    - Считает подпись: $s = y^e \mod N$
    - Отправляет (m, s)
3. Проверяющий:
    - Вычисляет $h(m)$

  - Вычисляет $w = s^d \mod N$
  - Проверяет: $h(m) = w$ → подпись подлинна

Безопасность:

- Хэш невозможно подделать без изменения сообщения.
- Подпись невозможно подделать без знания закрытого ключа.

---

4. Схема "подписать и зашифровать"

1. A подписывает сообщение: $s = m^d_A \mod n_A$
2. A шифрует s и m открытым ключом B: $C = s^{e_B} \mod n_B$
3. B расшифровывает своим закрытым ключом.
4. Проверяет подпись A с помощью e_A

Важно: Работает только если $n_A < n_B$, иначе возможна ошибка. Если это не так — используют два RSA-набора: для подписи и для шифрования.

🕶 5. Подпись "вслепую" (blind signature)

Позволяет получить подпись на сообщение, **не раскрывая его содержимого** отправителю.

Протокол:

1. Получатель выбирает случайное r, такое что $\gcd(r, n) = 1$
2. Формирует «заслеплённое» сообщение: $m' = r^e \cdot m \mod n$
3. Отправитель подписывает: $s' = (m')^d \mod n$
4. Получатель снимает «пелену»: $s = s' \cdot r^{-1} \mod n$

Подтверждение подлинности: Проверяется обычным способом: $s^e \mod n = m$

Применяется, например, в **анонимных голосованиях** и **электронных платежах**.

👤 6. Протокол аутентификации на базе RSA

Используется для подтверждения личности через криптографические методы.

Протокол:

1. Пользователь A публикует открытый ключ (e, m), закрытый d знает только он.

2. Сервер B посылает случайное число s.

3. A подписывает: $g = s^d \mod m$

4. B проверяет: $s' = g^e \mod m$

   - Если $s' = s$ — личность подтверждена.

Безопасность: Закрытый ключ не раскрывается, процесс устойчив к перехвату.

---

7. Закрытый обмен между двумя пользователями

Пользователи могут:

- Аутентифицироваться.
- Подписывать и шифровать сообщения.
- Шифровать даже **сами подписи**.

Для этого удобно иметь:

- По **3 пары ключей** (для подписи, шифрования, аутентификации).

Реализация обмена:

- Шифрование: открытым ключом получателя.
- Расшифровка: своим закрытым.
- Подпись: своим закрытым.
- Проверка подписи: открытым отправителя.

Примеры

Пример ручной подписи:

1. Участники выбирают простые числа и ключи.
2. Один шифрует, второй расшифровывает — если все ключи и модули подобраны правильно, получится исходное сообщение.

Пример в Maple:

Пошаговое вычисление ключей и подписей для больших чисел с использованием `numtheory`:

- `phi(n)` — функция Эйлера.
- `msolve` — решение сравнений.
- Проверка подписи и шифрование.

Закрепим

Понятие	Суть
Электронная подпись (RSA)	$s = x^d \mod m$, проверка: $s^e \mod m = x$
С хэшированием	Подписывается хэш от сообщения, а не само сообщение
Подпись + шифрование	Подпись закрытым ключом A, шифрование открытым ключом B
Подпись "вслепую"	Получатель скрывает сообщение, подписывающий его не видит
Аутентификация	Подпись случайной строки для подтверждения личности

---

08 RSA и криптоанализ: общий контекст

- RSA — асимметричная криптосистема с открытым (e, n) и закрытым (d) ключами.
- Основная идея криптоанализа — получение d или открытого текста M, зная только e, n, C.

Типы криптоаналитических атак

1. Ciphertext Only Attack (по зашифрованному тексту)

- Есть только C, n, e.
- Сложна, требует большого количества шифртекстов.
- Используется статистический анализ.

2. Known Plaintext Attack (по известному открытому тексту)

- Известны пары (M, C).
- Цель — вывести ключ или правила шифрования.

3. Chosen Plaintext Attack

- Аналитик сам выбирает M, получает C.
- Позволяет строить таблицы соответствий.

4. Adaptive Chosen Plaintext Attack

- Как выше, но выбор M меняется на основе полученных C.

5. Chosen Ciphertext Attack

- Аналитик выбирает C, получает M.
- Применима к RSA, особенно при наличии сервиса расшифровки.

6. Adaptive Chosen Ciphertext Attack

- Динамически выбирает C, анализируя результаты.

Конкретные атаки на RSA

1. Атака по открытому ключу (вычисление закрытого ключа)

- Если удалось разложить n = p*q, то:
  - Вычисляется φ(n) = (p-1)(q-1)
  - Решается уравнение e·d ≡ 1 (mod φ(n))

2. Атака угадыванием φ(n)

- Если известно φ(n), можно составить квадратное уравнение на p и q.

3. Метод Ферма

- Эффективен, если p и q близки: $n = x^2 - y^2 = (x - y)(x + y)$

4. Атака с общим модулем (вариант 1)

- Один и тот же n, разные e1, e2, одно и то же сообщение M.
- Используется расширенный алгоритм Евклида для нахождения r и s, далее: $M = (C1^r \cdot C2^s) \bmod n$

5. Атака с общим модулем (вариант 2)

- Повторное использование одного n с разными e.
- Тоже восстанавливается M через обобщённый алгоритм Евклида.

6. Атака с малой экспонентой (e = 3)

- Сообщение отправлено 3 получателям с разными n:
    - Используется китайская теорема об остатках
    - Извлекается кубический корень: $M = \sqrt[3]{C}$

7. Атака с малой экспонентой (другое описание)

- Аналогично выше, но упор на то, что $M^e < n1 n2 n3$.

8. Атака методом неподвижной точки (вариант 1)

- Повторное шифрование: если $C^k \bmod n = C$, то C — неподвижная точка.
- Тогда $M = C^{(k-1)} \bmod n$

9. Атака методом неподвижной точки (вариант 2)

- Определение неподвижной точки: $x^{(e^k)} \equiv x \pmod n$
- Позволяет извлечь M через степенные вычисления.

10. Атака на короткий открытый текст

- Если M мал, аналитик подбирает x, y, такие что: $C \cdot x^{(-e)} \equiv y^e \pmod n \rightarrow M = x \cdot y$

11. Атака Винера

- Работает, если $d < 1/3 \cdot n^{(1/4)}$
- Использует цепные дроби для приближения e/n и нахождения d.

Методы защиты:

- Не использовать малые d или e
- Добавлять «соль» и случайные данные перед шифрованием
- Избегать общего модуля n
- Не шифровать одно и то же сообщение разным e
- Увеличивать длину p и q и их различие

- Против квантовых атак — переход на постквантовую криптографию