УДК 519.61:003.26

ЧИСЛЕННЫЕ МЕТОДЫ В КРИПТОГРАФИИ: ОТ АЛГОРИТМОВ ЕВКЛИДА ДО RSA

Кардаш Д.О., Гук Н. И., Пархонюк М. П., студенты

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Степанова Т. С. – канд. физ.-мат. наук, доцент

Аннотация. В статье рассматривается математический фундамент современной асимметричной криптографии. Изложение строится от алгоритма Евклида и теоремы Ферма к системе RSA. Показана логическая цепочка: алгоритм деления → НОД → расширенный алгоритм Евклида → тест Миллера–Рабина → RSA. Для каждого метода приводятся формулировка, доказательство корректности и числовой пример.

Ключевые слова. алгоритм Евклида, НОД, теорема Ферма, тест Миллера–Рабина, RSA, асимметричная криптография, модулярная арифметика.

Введение. Криптография – область математики и информатики, обеспечивающая конфиденциальность, целостность и аутентичность информации. Наиболее известная асимметричная криптосистема – RSA, предложенная Ривестом, Шамиром и Адлеманом в 1978 году [1].

Как указано в учебном пособии [2], в основе построения RSA лежат два фундаментальных алгоритма теории чисел: алгоритм деления с остатком и алгоритм Евклида.

Безопасность RSA определяется вычислительной трудностью факторизации: для числа $n = p \cdot q$, где p и q – большие простые, разложить n на множители без знания этих простых практически невозможно [1]. Для построения таких систем необходимо уметь эффективно проверять простоту чисел и вычислять НОД – именно этому посвящена настоящая работа.

1. Алгоритм Евклида

1.1. Наибольший общий делитель

Определение 1.1 [1]. Целое b делит целое a, если существует целое c такое, что $a = b \cdot c$. Наибольший общий делитель $d = \text{НОД}(a, b)$ – наибольшее целое, делящее оба числа. Числа взаимно просты, если $\text{НОД}(a, b) = 1$.

Лемма 1.1 [1]. Пусть $a = b \cdot q + r$. Тогда $\text{НОД}(a, b) = \text{НОД}(b, r)$. Доказательство: если d делит a и b, то делит и $r = a - b \cdot q$, и наоборот. Следовательно, множества общих делителей пар (a, b) и (b, r) совпадают, откуда их максимумы равны.

1.2. Классический алгоритм Евклида

Алгоритм 1.1 (алгоритм Евклида) [1].
Ввод: натуральные $a > b$.
Вывод: $\text{НОД}(a, b)$.
Шаг 1. $A = a$; $B = b$.
Шаг 2. $R = A \bmod B$.
Шаг 3. Если $R = 0$: вернуть B..
Шаг 4. $A = B$; $B = R$; перейти к шагу 2.

Теорема 1.1 (алгоритм Евклида) [1]. Последний ненулевой остаток равен $\text{НОД}(a, b)$. Конечность алгоритма обусловлена строгим убыванием остатков: $0 \le \ldots < r_3 < r_2 < r_1 < b$. Каждая итерация сохраняет НОД по лемме 1.1.

Таблица 1 – Выполнение алгоритма Евклида для НОД(1234, 54)

Шаг	a	b	r = a mod b	НОД
1	1234	54	46	= НОД(54, 46)
2	54	46	8	= НОД(46, 8)
3	46	8	6	= НОД(8, 6)
4	8	6	2	= НОД(6, 2)
5	6	2	0	= 2

Результат: НОД(1234, 54) = 2. Число итераций алгоритма O(log min(a, b)). По теореме Ламе [1]: число шагов не превышает пятикратного числа десятичных цифр меньшего из чисел.

1.3. Расширенный алгоритм Евклида

Теорема 1.2 (тождество Безу) [1]. Для любых натуральных a и b существуют целые α, β такие, что α·a + β·b = НОД(a, b). Коэффициенты Безу вычисляются рекуррентным дополнением таблицы Евклида: $x_n = x_{n-2} - q_n \cdot x_{n-1}$, $y_n = y_{n-2} - q_n \cdot y_{n-1}$, $x_{-1} = 1$, $y_{-1} = 0$, $x_0 = 0$, $y_0 = 1$.

Пример 1.1. Для НОД(1234, 54): α = −7, β = 160. Проверка: (−7)·1234 + 160·54 = −8638 + 8640 = 2. Применение в RSA: секретный ключ $d = e^{-1} \bmod \varphi(n)$ вычисляется запуском расширенного алгоритма Евклида для пары (e, φ(n)).

2. Малая теорема Ферма

2.1. Арифметика остатков

Кольцо вычетов $\mathbb{Z}_n = \{0, 1, ..., n-1\}$ с операциями mod n. По теореме обратимости [1]: элемент ā обратим в $\mathbb{Z}_n$ тогда и только тогда, когда НОД(a, n) = 1. Обратный элемент $a^{-1} \bmod n$ вычисляется расширенным алгоритмом Евклида. Это ключевое свойство для RSA: открытая экспонента e выбирается взаимно простой с φ(n), что гарантирует существование $d = e^{-1} \bmod \varphi(n)$.

2.2. Формулировка и доказательство

Теорема 2.1 (малая теорема Ферма) [1]. Пусть p — простое, a — любое целое. Тогда $a^p \equiv a \pmod p$. Если $p \nmid a$, то $a^{p-1} \equiv 1 \pmod p$.

Доказательство проводится индукцией по a. База: $1^p = 1$. Шаг: пусть $n^p \equiv n \pmod p$. По лемме о биноме в $\mathbb{Z}_p$: $(n+1)^p \equiv n^p + 1^p \equiv n + 1 \pmod p$, поскольку биномиальный коэффициент C_p^k при $0 < k < p$ кратен p.

Следствие позволяет редуцировать большие показатели: $a^k \equiv a^r \pmod p$, где $r = k \bmod (p-1)$. Пример 2.1 [1]: $2^{5\,4\,3\,2\,6\,7\,5} \bmod 13$. Так как $2^{12} \equiv 1 \pmod{13}$ и 5432675 = 12·452722 + 11, получаем $2^{5\,4\,3\,2\,6\,7\,5} \equiv 2^{11} = 2048 \equiv 7 \pmod{13}$.

3. Тест Миллера–Рабина

3.1. Псевдопростые числа и числа Кармайкла

Определение 3.1 [1]. Нечётное составное n называется числом Кармайкла, если $b^{n-1} \equiv 1 \pmod n$ при НОД(b, n) = 1. Простейший пример: 561 = 3·11·17. Теорема Корселта [1]: нечётное n – число Кармайкла тогда и только тогда, когда для каждого простого делителя p: $p^2 \nmid n$ и $(p-1) \mid (n-1)$. Числа Кармайкла обходят тест Ферма, поэтому требуется более сильный тест.

3.2. Алгоритм теста

Основная идея [1]: записать $n-1 = 2^k \cdot q$, q нечётно. Если n простое, то для основания b в последовательности $b^q, b^{2q}, ..., b^{2^k q}$ либо первый элемент ≡ 1, либо найдётся элемент ≡ n−1 (mod n). Нарушение доказывает составность n.

Алгоритм 3.1 (тест Миллера–Рабина) [1].

Вход: нечётное n > 0, основание b: 1 < b < n−1.

Шаг 1. Найти k, q: $n-1 = 2^k \cdot q$, q нечётно.

Шаг 2. $r = b^q \bmod n$; i = 0.

Шаг 3. Если (i = 0 и r = 1) или r = n−1: «вероятно простое» – стоп.

Шаг 4. i = i+1; $r = r^2 \bmod n$.

Шаг 5. Если i < k: перейти к шагу 3. Иначе: «n составное».

Теорема Рабина [1]: при k случайных основаниях вероятность ошибки $\le 4^{-k}$. При k = 40 – это менее 10^{-24} . Тест является промышленным стандартом генерации простых чисел для RSA.

Таблица 2 – Тест Миллера для n = 561, b = 2 ($560 = 2^4 \cdot 35$)

$2^{35} \bmod 561$	$2^{70} \bmod 561$	$2^{140} \bmod 561$	$2^{280} \bmod 561$
263	166	67	1

Вывод: 561 составное, так как среди предшествующих единице элементов последовательности нет значения n−1 = 560. Тест не обманывается числом Кармайкла.

4. Алгоритм RSA

4.1. Математическое обоснование

Теорема Эйлера [3]. Для НОД(M, n) = 1: $M^{\varphi(n)} \equiv 1 \pmod n$. Если $e \cdot d \equiv 1 \pmod{\varphi(n)}$, то $(M^e)^d = M^{e \cdot d} = M^{k \cdot \varphi(n)+1} \equiv M \pmod n$. Это гарантирует корректность дешифрования: зашифровав открытым ключом, получим исходное сообщение при применении секретного.

4.2. Генерация ключей

1. Выбрать простые p и q (≥ 512 бит) с помощью теста Миллера–Рабина.
2. n = p·q – модуль RSA; φ(n) = (p−1)·(q−1) – функция Эйлера.
3. Выбрать e: 1 < e < φ(n), НОД(e, φ(n)) = 1.

Наиболее распространённое значение открытой экспоненты e = 65537 = 2^{16} + 1. Оно является простым, имеет малый вес Хэмминга (две единицы в двоичной записи), что ускоряет возведение в степень. Меньшие значения (например, e = 3) подвержены атакам при малом сообщении.

4. $d = e^{-1} \bmod \varphi(n)$ — расширенным алгоритмом Евклида.

5. Открытый ключ (e, n). Секретный ключ (d, n). p, q, φ(n) уничтожить.

Современные рекомендации (NIST SP 800-57) требуют для RSA длину модуля n не менее 2048 бит для обеспечения стойкости до 2030 года. При этом p и q выбираются примерно одинаковой длины — около 1024 бит каждое.

4.3. Числовой пример

Таблица 3. Пошаговый пример RSA (p = 61, q = 53)

Параметр	Значение	Пояснение
p	61	1-е простое
q	53	2-е простое
$n = p \cdot q$	3233	Модуль RSA (открытый)
φ(n)	3120	Функция Эйлера (секрет)
e	17	НОД(17, 3120) = 1
$d = e^{-1} \bmod \varphi(n)$	2753	$17 \cdot 2753 = 46801 = 15 \cdot 3120 + 1$
Открытый ключ	(17, 3233)	Публикуется
Секретный ключ	(2753, 3233)	Хранится в тайне
Сообщение M	65	Числовое представление
Шифртекст $C = M^e \bmod n$	2790	$65^{17} \bmod 3233$
Расшифровка $M = C^d \bmod n$	65	$2790^{2753} \bmod 3233 = 65$

Ключевой шаг – нахождение d: применяем расширенный алгоритм Евклида к паре (17, 3120). Проверка: $17 \cdot 2753 = 46801 = 15 \cdot 3120 + 1 \equiv 1 \pmod{3120}$.

4.4. Быстрое возведение в степень

При практическом применении RSA необходимо вычислять выражения вида $C = M^e \bmod n$ и $M = C^d \bmod n$, где показатель степени достигает 1024 бит и более. Прямолинейное последовательное умножение, требующее порядка e операций, при таких значениях показателя становится вычислительно неосуществимым: число шагов составило бы около 2^{1024}, что многократно превышает возможности любых современных вычислительных систем.

Эффективное решение основано на разложении показателя по степеням двойки. Запишем e в двоичном виде: $e = b_m \cdot 2^m + b_{m-1} \cdot 2^{m-1} + \ldots + b_1 \cdot 2 + b_0$, где $b_i \in \{0, 1\}$, b_m – старший бит. Тогда M^e представляется как произведение множителей вида $(M^{2^i})^{b_i}$, причём каждый множитель M^{2^i} получается из предыдущего одним возведением в квадрат. Это наблюдение лежит в основе метода бинарного возведения в степень.

Наиболее распространённым вариантом является обход битов показателя от старшего к младшему (метод MSB-first, или «слева направо»). Алгоритм поддерживает промежуточный результат R, инициализируемый единицей; на каждом шаге он возводится в квадрат, а при обнаружении единичного бита дополнительно умножается на основание M. После обработки всех битов в R накапливается искомое значение.

Алгоритм 4.4 (бинарное возведение в степень, обход слева направо) [4].

Вход*:* основание M, показатель $e = (b_m\, b_{m-1} \ldots b_1\, b_0)_2$, модуль n.

Выход*:* $R = M^e \bmod n$.

Шаг 1. Положить R = 1.

Шаг 2. Для i от m до 0 (от старшего бита к младшему):

1) выполнить $R = R^2 \bmod n$;
2) если $b_i = 1$, выполнить $R = R \cdot M \bmod n$.

Шаг 3. Вернуть R.

Докажем, что по завершении итерации с индексом i промежуточный результат R содержит значение M^{e_i}, где e_i – число, образованное старшими битами e начиная с b_m до b_i включительно. При i = m после первого шага $R = M^{b_m}$, что соответствует инварианту. На каждой последующей итерации

возведение в квадрат сдвигает показатель влево (умножает на 2), а умножение на M добавляет единицу в показатель, если текущий бит равен 1. Таким образом, после обработки бита b_0 в R сосредоточено значение $M^e \bmod n$.

Алгоритм выполняет ровно (m + 1) возведений в квадрат и не более (m + 1) умножений, то есть не более 2(m + 1) = O(log e) модульных операций. При $e \approx 2^{1024}$ это порядка 1024, что на много порядков меньше прямолинейного подхода. Именно благодаря алгоритму 4.4 RSA остаётся практически применимым: шифрование и генерация подписи выполняются за миллисекунды даже при 2048-битном модуле.

Заключение. Алгоритм Евклида и его расширенная версия обеспечивают эффективное вычисление наибольшего общего делителя и обратного элемента в кольце вычетов, что необходимо для нахождения секретной экспоненты $d = e^{-1} \bmod \varphi(n)$. Малая теорема Ферма объясняет корректность схемы шифрования и дешифрования: возведение сообщения в степень e, а затем в степень d даёт исходное значение по модулю n. Тест Миллера–Рабина позволяет с пренебрежимо малой вероятностью ошибки удостовериться в простоте чисел p и q, из которых строится модуль RSA. Алгоритм бинарного возведения в степень сводит число модульных умножений к O(log e), делая шифрование и дешифрование практически осуществимыми при криптографически значимых длинах ключей.

Список использованных источников:

1. Rivest R., Shamir A., Adleman L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems // Comm. ACM. — 1978.

2. Ильин М.Е., Ципоркова К.А. Теоретико-числовые методы в криптографии. Ч. 1: учеб. пособие. — Рязань: РГРТУ, 2020. — 112 с.

3. Кнут Д.Э. Искусство программирования. Т. 2: Получисленные алгоритмы. — М.: Вильямс, 2007. — 832 с.

4. Шнайер Б. Прикладная криптография. — М.: Триумф, 2002. — 816 с.

5. Menezes A., van Oorschot P., Vanstone S. Handbook of Applied Cryptography. — CRC Press, 1996. — 780 p.

UDC 519.61:003.26

NUMERICAL METHODS IN CRYPTOGRAPHY: FROM EULER ALGORITHMS TO RSA

Kardash. D. O., Huk N. I., Parkhaniuk M. P., students

Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus

Stepanova T. S. – Candidate of Physical and Mathematical Sciences

Abstract. The article discusses the mathematical foundation of modern asymmetric cryptography. The presentation is based on the Euclidean algorithm and Fermat's theorem, leading to the RSA system. A logical chain is presented: division algorithm → greatest common divisor → extended Euclidean algorithm → Miller–Rabin test → RSA. Each method is described, including its formulation, proof of correctness, and numerical example.

Keywords. Euclidean algorithm, greatest common divisor, Fermat's theorem, Miller–Rabin test, RSA, Euler function, asymmetric cryptography, modular arithmetic.