Бинарный алгоритм возведения в степень и его прикладное значение

Приньков А. С.

Приньков Алексей Сергеевич / Prinkov Alexey Sergeevich – студент,
кафедра прикладной математики,
факультет автоматизации и информатики,
Липецкий государственный технический университет, г. Липецк

Аннотация: *в статье анализируется эффективность бинарного алгоритма возведения в степень, и необходимость его применения на практике для экономии ресурсов и стабильной работы криптосистем. Выявлена обобщенность алгоритма и возможность его применения для различных алгебраических структур.*

Ключевые слова: *бинарный алгоритм в схеме Эль - Гамаля, эффективность алгоритма, возведение в степень, математика криптографии.*

Возведение в степень – бинарная операция, определенная как результат многократного выполнения операции над аргументом: $a^b = \underbrace{a \bullet \ldots \bullet a}_{b}$, где $\bullet$ - некоторая операция. Это определение справедливо для объектов и операций различной природы. Для удобства интерпретации и программной реализации, рассмотрим полную мультипликативную группу кольца вычетов.

Прежде всего, необходимо установить критерии эффективности алгоритма. Мы будем рассматривать быстродействие как меру эффективности [1].

$t = c \times k$, где t – общее время работы, c – число операций, k – потребность во времени на одну операцию.

$t = \sum_i c_i \times r_i$ - формула для операций с разной потребностью времени.

Сравним два алгоритма: по определению и бинарный. Если $x = \left(x_n x_{n-1} \ldots x_0\right)_{10}$, то для вычисления x^b потребуется n^b операций (умножение цифр каждой на каждую) [2]. Данный подход, очевидно является неэффективным. Сравнение по модулю следует производить по мере возведения, а именно b – раз.

Далее разберем бинарный алгоритм. Пусть нужно найти x^k, где $x, k \in \mathbb{Z}_p$. Для этого найдем двоичное представление $k = \left(m_n m_{n-1} \ldots m_0\right)_2$, тогда $x^k = x^{((\ldots((m_n + m_{n-1})\times 2 + m_{n-2})\times 2 + \ldots)\times 2 + m_1)\times 2 + m_0} = ((\ldots((x^{m_k})^2 \times x^{m_{k-1}})^2 \ldots)^2 \times x^{m_0}$. Этот способ, требует $\frac{1}{2}\ln(n)$ операций [3]. Сравнение по модулю, после каждого возведения в квадрат поможет сэкономить память для хранения промежуточного результата. Иначе можно сравнить конечный результат.

За счет двоичности компьютеров, бинарный метод нашел обширное применение в криптосистемах с открытым ключом[4].

Обобщенный бинарный алгоритм, $t = x^k$:

1. Найти двоичное представление степени k
2. Положить объекты q = x, t = x
3. От $i = 0$ до $k - 1$ включительно выполнить
4. Если $m_i = 1$, то t = t.multiply(z) и z = z.multiply(z);

иначе z = z.multiply(z);

//где multiply - операция умножения

Для эмпирической оценки преимущества алгоритма сравним скорость работы двух вышеописанных подходов в схеме Эль-Гамаля, при поточном шифровании данных.

Таблица 1. Технические характеристики ЭВМ

Процессор	CPU CryptoHash	CPU Fibonacci	CPU Blowfish
4 x intel Core i5-4210U 2.7 GHz	287	1.82	4.71

Сравнение (не менее 400 запусков для каждого параметра, с различными значениями):

Основание $\mathbb{Z}_p$, значащих цифр	Бинарный алгоритм, секунды	Алгоритм по определению, секунды
10	0.1 - 0.15	0.14 - 0.22
100	0.120 – 0.190	0.92 – 1.4
1000	0.3 - 0.6	12 – 35
10000	6 - 15	>60

Никакой алгоритм нельзя назвать, окончательно, самым эффективным [1]. Поскольку в роли платформы для шифрования может выступать не только кольцо вычетов, а структура любой природы, отвечающая требованиям криптосистемы, то появляется потребность в алгоритме, который будет работать с разными структурами, не теряя своей эффективности. Бинарный алгоритм сочетает эффективность и абстрактность, поэтому является полезным и стабильным решением для криптографии.

Литература

1. *Разборов А. А*. О сложности вычислений // Математическое просвещение. 1999. № 3. С. 127-141
2. *Панкратова И. А.* Теоретико – числовые методы криптографии. Томск: Томский государственный университет. 2009. 120 с.
3. *Шнайер Б*. Алгоритмы с открытыми ключами // Прикладная криптография. М.: Триумф. 2002. 610 с.
4. *Коутинхо* С. Введение в теорию чисел. Алгоритм RSA. М.: Постмаркет. 2001. 328 с.