Завгородний Станислав Дмитриевич

студент

ФГАОУ ВО «Самарский национальный исследовательский университет им. академика С.П. Королева»

г. Самара, Самарская область

ДИСКРЕТНОЕ ЛОГАРИФМИРОВАНИЕ. АЛГОРИТМ БОЛЬШИХ И МАЛЫХ ШАГОВ

Аннотация: *в статье рассматривается алгоритм больших и малых шагов. В основе работы лежат понятия о дискретном логарифмировании, группах, кольцах и классах вычетов. Автором также реализуется программа дискретного логарифмирования.*

Ключевые слова: *дискретное логарифмирование, группа, кольцо, класс вычетов.*

Основные понятия

Введем некоторые определения, которые будут использованы в работе.

Определение 1. Дискретное логарифмирование – это задача обращения функции g^x в некоторой мультипликативной группе G.

Определение 2. Группа – множество, на котором определена ассоциативная бинарная операция, для которой имеется нейтральный элемент и каждый элемент множества имеет обратный. Одним из примеров группы является множество целых чисел.

Определение 3. Кольцо – это алгебраическая структура, в которой определены операция обратимого сложения и умножения.

Определение 4. Класс вычетов a по модулю m – это множество всех чисел, сравнимых с a по модулю m.

Введение

Так же ,как и задача разложения на множители, задача дискретного логарифмирования находит применение во многих криптографических алгоритмах с

открытым ключом. В 1976 году У. Диффи и М. Хеллман предложили задачу дискретного логарифмирования для установления сеансового ключа, она послужила основой для создания протоколов цифровой подписи и шифрования и других криптографических протоколов.

Дискретное логарифмирование

Решение уравнения $g^x \equiv h$ для заданных g и h называется дискретным логарифмом элемента h по основанию g. Если *G* является мультипликативной группой кольца вычетов по модулю *m,* то решение можно также назвать индексом числа *a* по основанию *g*. Индекс числа a по основанию g обязательно будет существовать, если g является первообразным корнем по модулю *m*. Пусть задано в некоторой конечной мультипликативной группе уравнение $g^x \equiv h$ (1). Решение задачи дискретного логарифмирования состоит в нахождении некоторого целого неотрицательного числа, удовлетворяющего уравнению (1). Если оно разрешимо, у него будет хотя бы одно натуральное решение, которое не превышает порядок группы. Это даёт первичную оценку сложности алгоритма поиска решений сверху. Алгоритм «грубой силы» отыскал бы решение за число шагов не выше порядка данной группы.

Обычно рассматривается случай, когда группа циклическая, тогда уравнение всегда имеет решение. Если группа произвольная, то вопрос о разрешимости задачи дискретного логарифмирования требует отдельного рассмотрения.

Таблица 1

Пример дискретного логарифма $3^x \equiv 13 \pmod{17}$. X = 4

$3^1 \equiv 3$	$3^2 \equiv 9$	$3^3 \equiv 10$	$3^4 \equiv 13$	$3^5 \equiv 5$	$3^6 \equiv 15$	$3^7 \equiv 11$	$3^8 \equiv 16$
$3^9 \equiv 14$	$3^{10} \equiv 8$	$3^{11} \equiv 7$	$3^{12} \equiv 4$	$3^{13} \equiv 12$	$3^{14} \equiv 2$	$3^{15} \equiv 6$	$3^{16} \equiv 1$

Алгоритм больших и малых шагов

Пусть мы имеем уравнение: $a^x = b \pmod{m}$, где a и m взаимно просты.

Преобразуем уравнение, обозначив x = np – q, где n – это выбранная константа. P обычно называют «giant step», так как увеличение p на единицу увеличивает x на n, а q соответственно называют «baby step».

Любое x из промежутка [0; m) можно представить в данной форме, при этом будет достаточно значений: p принадлежит [1; [m / n]], где q принадлежит [1; n].

Уравнение примет вид: $a^{np-q} = b \pmod m$, а так как a и m взаимно просты, получаем: $a^{np} = ba^{q} \pmod m$. Чтобы решить это уравнение, нужно найти соответствующие значения p и q, чтобы значения левой и правой частей совпали, иными словами нужно решить уравнение: $f_1(p) = f_2(q)$.

Данная задача решается с помощью метода «встреча по середине». Первая часть алгоритма: считаются значения функции f_1 для всех значений аргумента p и после этого эти значения сортируются. Вторая часть алгоритма: перебираются значение второй переменной q и вычисляется вторая функция f_2, и ищется это значение среди предвычисленных значений функции f_1 с помощью двоичного поиска. Время работы такого алгоритма оценивается как O (sqrt (n)), а это намного лучше, чем время работы алгоритма «грубой силы» показателей степени O (n).

Реализации программы вычисления дискретного логарифма

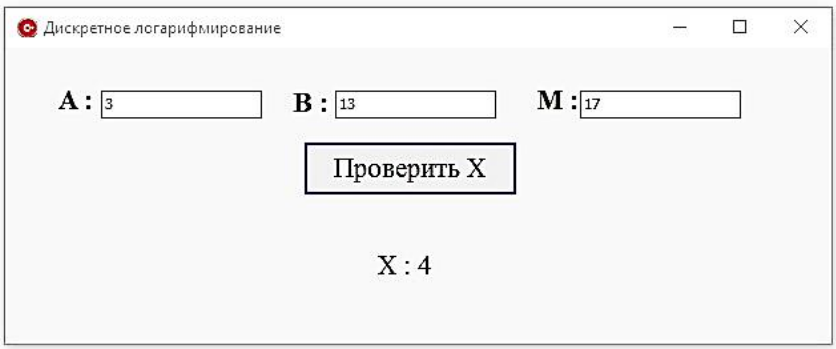

Рис. 1. Пример работы разработанной программы

На основе алгоритма больших и малых шагов на языке C++ в среде разработок C++ Builder была реализована программа вычисления дискретного логарифма. В качестве примера приведено окно ввода начальных значений и вывода результата вычисления программы. Входные значения A = 3, B = 13, M = 17, результат вычисления X = 4.

Список литературы

1. Нестеренко Ю.В. Глава 4.8. Дискретное логарифмирование // Введение в криптографию / Под ред. В.В. Ященко. – СПб.: Питер, 2001. – 288 с.

2. Василенко О.Н. Теоретико-числовые алгоритмы в криптографии. – М.: МЦНМО, 2003. – 328 с.