Тестирование критических случаев работы алгоритмов. Построение блок-схемы работы алгоритмов.

В криптографии под случайным простым числом понимается простое число, содержащее в двоичной записи заданное количество битов k, на алгоритм генерации которого накладываются определенные ограничения. Получение случайных простых чисел является неотъемлемой частью процедур выработки ключей во многих криптографических алгоритмах, включая RSA и ElGamal.

Ввиду того, что тестирование простоты больших чисел требует существенных временных затрат, требование простоты получаемого числа часто ослабляют до сильной псевдопростоты по нескольким различным случайным основаниям. Существующие алгоритмы тестирования сильной псевдопростоты на порядки быстрее лучших известных алгоритмов тестирования простоты. В то же время, числа, успешно прошедшие тестирования сильной псевдопростоты по нескольким случайным основаниям, с большой вероятностью являются простыми, причем эта вероятность растет с ростом количества оснований, по которым проводится тестирование.

Требования к алгоритму и его реализации

Требования к алгоритмам генерации случайных простых чисел сводятся к следующим двум:

Распределение получаемых простых чисел должно быть близко к *равномерному* на множестве всех простых чисел, содержащих k битов.

Существует несколько способов обеспечить выполнимость этого требования.

Процесс генерации конкретного случайного простого числа нельзя воспроизвести, даже зная детали алгоритма и его реализации. Обычно выполнение этого требования обеспечивается использованием криптостойкого ГПСЧ, проинициализированного некоторым ключом, получаемым извне (т. е. не

являющимся частью алгоритма или его реализации). В качестве ключа может выступать, например, значение криптостойкой хэш-функции от секретной фразы, запрашиваемой у пользователя.

Тестирование простоты

Проверить (вероятную) простоту числа p, содержащее k битов, можно так:

Убедиться, что p не делится на небольшие простые числа 3, 5, 7, 11, и т.д. до некоторого небольшого предела (например, 256). Такая проверка позволяет эффективно отсечь множество заведомо составных чисел, прежде чем проверять их посредством более трудоёмких алгоритмов. Так, проверка делимости p на простые числа 2, 3, 5 и 7 отсеивает все четные числа и 54% нечетных чисел, проверка делимости p на все простые числа до 100 отсеивает 76% нечетных чисел, а проверка делимости p на все простые числа до 256 отсеивает 80% нечетных чисел.

Выполнить тест Миллера — Рабина с количеством раундов не меньше k.

Если число p не проходит хотя бы одной проверки — оно не является простым. В противном случае с большой вероятностью (зависящей от количества раундов) число p является простым.

Типовой алгоритм 1

Сгенерировать k − 1 случайных битов и составить из них k-битное число p (старший бит равен 1).

Увеличить p на 1 и проверить его простоту. Повторять этот шаг до тех пор, пока не будет найдено простое число.

Второй шаг можно ускорить, если рассматривать только нечетные числа или числа сравнимые с 1 и 5 по модулю 6 и т.п.