УДК 004.056.55

Алгоритм AES для шифрования государственной тайны

Замковец В.В., Несенчук А.А.
Белорусский национальный технический университет

В настоящее время для шифрования государственной тайны в разных государствах используются различные стандарты и алгоритмы. Так, в СНГ применяется алгоритм шифрования ГОСТ 28147-89. В Евросоюзе стандартизация алгоритмов шифрования осуществляется в рамках проекта NESSIE, в Японии аналогичную роль выполняет организация CRYPTREC.

Государственным стандартом шифрования США является Advanced Encryption Standard (AES) [1]. Стандарт AES основан на алгоритме блочного шифрования Rijndael, который был отобран среди многих других алгоритмов в конкурсе, организованном Национальным институтом стандартов и технологий (NIST) США в 1997 г.

В работе выполнена программная реализация данного алгоритма на языке C#. Алгоритм обрабатывает шифруемые данные в виде блоков длиной 128 бит. Он принимает секретный ключ, длина которого может быть выбрана равной одному из значений – 128, 192 или 256 бит. Данные при обработке представляются в виде матрицы 4x4 байта. После процедуры расширения ключа к каждому байту блока шифруемых данных и каждому байту полученного значения ключа применяется операция XOR (исключающее «ИЛИ»). Затем каждый байт заменяется соответствующим значением в специальной фиксированной таблице значений. Далее в матрице происходит сдвиг значений по строкам влево на количество байтов, равное индексу строки. Затем происходит умножение каждого столбца матрицы (в виде полиномов) в поле Галуа по модулю $x^4 + 1$ на фиксированный многочлен $c(x) = 3x^3 + x^2 + x + 2$. Результат вновь смешивается с ключом путём применения операции XOR. Данные действия повторяются несколько раз в зависимости от выбранного числа раундов. При дешифровании с тем же ключом все действия выполняются в обратном порядке.

В направлении усовершенствования в алгоритм введена процедура внесения дополнительной диффузии в шифр. Суть модернизации состоит в замене двумерной матрицы трёхмерной. Длина блоков данных в этом случае равна 512 бит. Сдвиг байтов в матрице проводится вначале вышеупомянутым способом – по столбцам во втором измерении, а затем аналогичным образом – по столбцам в третьем измерении. Такое новшество подходит для шифрования больших объёмов данных и усиливает алгоритм.

Литература:

1. Daemen J., Rijmen V. The Design of Rijndael: AES – Advanced Encryp-

tion Standard. – N.Y.: Springer, 2002.

УДК 004.421.2

Рекурсивное программирование

Ковальков А.Т.
Белорусский национальный технический университет

Рекурсия, т.е. обращение некоторой подпрограммы (процедуры или функции) к самой себе, имеется практически во всех современных языках программирования. Отношение программистов к рекурсии неоднозначно – от восторженных отзывов до почти полного отрицания. В целом использование рекурсии даже опытными программистами не так активно, как она этого заслуживает. Между тем рекурсия является фундаментом таких языков программирования, как Пролог и Лисп, в которых она стала основным механизмом программирования.

Сдержанность многих программистов к рекурсии объясняется главным образом тем, что они, привыкшие при программировании на процедурном языке разрабатывать алгоритм решения задачи, с этих же позиций программируют и рекурсивную процедуру, стараясь вникнуть в достаточно непростой механизм реализации рекурсии. Такой подход непродуктивен.

Есть другой метод рекурсивного программирования, суть которого в следующем. Конец рекурсивных вызовов определяется граничным или терминальным условием, которое строится заданием таких значений входным параметрам, для которых сразу можно записать значения выходных параметров. При построении рекурсивных предложений предполагаем, что рекурсивный вызов процедуры или функции при измененных входных параметрах вычисляет промежуточный результат, который на один шаг отличается от окончательного результата. Решение получаем, используя промежуточный результат. Такой подход к программированию позволяет просто, без разработки алгоритма и не вникая во внутренний механизм работы рекурсии, логически конструировать нужные рекурсивные подпрограммы.

Таким образом, использование предлагаемой опробованной на практике методики построения рекурсивных подпрограмм позволяет просто использовать рекурсию не только в декларативных языках программирования, к которым относятся такие языки как Пролог и Лисп, но и в процедурных языках типа Паскаль, Си.

Построение рекурсивных подпрограмм по предлагаемой методике позволяет превратить программирование и на процедурных, и на декларативных языках в увлекательное занятие, равноценное решению